thalescpl.com

THALES

Live Data Transformation Guide

6.3.0

Document version 1

March 31, 2020

Live Data Transformation Guide
6.3.0
Document version 1
March 31, 2020

NOTICES, LICENSES, AND USE RESTRICTIONS

Contents

Preface

The Live Data Transformation Guide describes how to configure and use Live Data Transformation.

Live Data Transformation (LDT) is an optional, separately licensed feature of Vormetric Transparent Encryption (VTE). With LDT, DSMGDE Appliance administrators can change the encryption key and re-encrypt GuardPoint data without suspending user or application access to the data.

Intended Audience

The VTE Data Transformation Guide is for security teams who want to rekey the existing GuardPoint data, or who need to perform an initial encryption of their GuardPoint data.

Assumptions

Live Data Transformation is an enhancement to existing functionality. This documentation assumes the reader is familiar with the following Vormetric products and processes:

- Vormetric Data Security Manager (DSMGDE Appliance)
- Vormetric Transparent Encryption (VTE)
- Key management
- Data encryption

Sales and Support

For support and troubleshooting issues:

- https://supportportal.thalesgroup.com
- (800) 545-6608

For Thales Sales:

- https://enterprise-encryption.vormetric.com/contact-sales.html
- sales@thalesesecurity.com
- (888) 267-3732

Chapter 1: Introduction to Live Data Transformation

This chapter introduces Live Data Transformation (LDT), provides a high-level overview of how it works, and compares using VTE with and without LDT.

It contains the following sections:

Overview of Live Data Transformation

Live Data Transformation (LDT) is an optional, separately licensed feature of Vormetric Transparent Encryption (VTE). With LDT, after enabling a GuardPoint, DSM administrators can encrypt, or rekey, GuardPoint data without blocking user or application access to the data. In LDT, **rekey** means decrypting data with the current cryptographic key and then encrypting it with a new cryptographic key. The concept of rekey, and how LDT rekeys data, is described in this document.

After enabling GuardPoints, LDT performs **initial** encryption or rekeying in the background, unnoticed by users. The data stays live and available. This accelerates VTE deployments and eliminates the need to block application and user access to data during encryption or rekey operations, which can seriously inconvenience users and affect operational efficiency.

With LDT, the DSM administrator can create a single Live Data Transformation policy for both initial encryption and subsequent rekeying. The same policy applies to production access and security rules without restricting user or application access to data. Applications have continuity of access to GuardPoint data during LDT.

Using VTE without LDT

In contrast, when you use VTE without the LDT feature, access to GuardPoint data is unavailable when the data is encrypted during the initial data transformation with a standard policy with data transformation rule(s). Whenever the encryption key changes, access to data becomes unavailable again so the data can be re-encrypted with the new key under another standard policy with data transformation rule(s). After the initial encryption, or rekey operation completes, you must remove the GuardPoint from the transformation policy. Then you must guard again with a separate policy that uses the new key and allows access to the data in the GuardPoint.

WARNING

To prevent data loss or corruption, you must stop all applications and users that are accessing files inside a GuardPoint directory before enabling an LDT policy for that directory. This is the same requirement imposed with non-LDT policies. Terminating applications closes files accessed inside a GuardPoint before a policy is applied to the GuardPoint directory. Users can restart applications after the DSM administrator applies the LDT policy to the GuardPoint. This is the only downtime in application service imposed by LDT policies.

Use Cases

This section provides a summary of typical uses for LDT. The concepts mentioned in this section are described in more detail throughout the rest of this guide.

1. Encrypt unprotected data

   When protecting files in a directory, you must encrypt them. This process is called *initial data encryption*.
2. Convert non-LDT GuardPoints to LDT GuardPoints

   Use when you have existing GuardPoints that are protected with policies created before you started using LDT.
3. Rekey process

   Changing the key from one version to another version of the same key provides more security. Using LDT, you can change the encryption keys to more secure keys.
4. Transform the encrypted data to clear data.

Keys in LDT (Versioned Keys)

LDT uses **versioned keys.** When you create a versioned key, you set its **life span**. The life span is the duration (in days) of the versioned key.

You also define an **initial expiration date**. When the key reaches its expiration date, it automatically rotates to a new version. Then, it rotates every time its life span ends.

Key rotation is the creation of new cryptographic material for a key. It uses the same encryption algorithm and key name that were originally defined when the key was created. You can use the same key name for multiple transformations over time, because LDT creates a new version of the same key periodically. LDT uses the new key material to transform data to the new key version, as part of the same LDT policy that also protects data.

If you do not use LDT, you have to create multiple unique keys and write separate data transformation policies: one policy for initial encryption and a different policy for subsequent production usage. If you want to change the key, you must create a new key and a new policy to go with it. The LDT technique, using a single policy with a versioned key, is more convenient to configure.

Versioned keys all have the same properties as non-versioned keys, plus three additional properties:

- **Automatic Key Rotation**:

  Determines whether the key automatically rotates. For use with LDT, this property is always selected. Selection of this property implies a versioned key.
- **Key Version Life Span (days)**:

  Frequency of key rotation in days. Applies only if Automatic Key Rotation is selected.
  - “Converting a Non-LDT GuardPoint to an LDT GuardPoint” on page 40
- **Expiration Date**:

  LDT uses the initial version of the key until this date occurs. On the expiration date, LDT creates a new key version. Subsequently, it creates new versions based on the Key Version Life Span.

WARNING

You **must** specify the **Expiration Date**. Without an Expiration Date, a versioned key never rotates from the initial key version, and consequently, never triggers live data transformation.

Rekey | Key rotation

In LDT, **rekeying** means decrypting the data with a previous version of the key, and re-encrypting it with a new version of the key. LDT allows users and applications to access data while LDT is rekeying the data. Rotating the key and re-encrypting the GuardPoint data with the new version of the key helps to maintain a high level of data security.

Most often, the rekey happens automatically, because each versioned key has a Key Version Life Span that specifies the lifespan of the key. In addition to this automatic key version rotation, you can manually generate a new version of the current key if a new version is required.

More information:

- "Creating a New Versioned Key for LDT" on page 23
- "Rotating Encryption Keys (Rekey)" on page 41

Live Data Transformation Policies

In LDT, you define a single policy for initial data encryption and subsequent rekeying. The policy specifies:

- **Current key**

  Associated with data that you want to protect using Live Data Transformation. This is either a non-versioned key from an earlier policy, or `clear_key`, which means that the data is not currently encrypted.
- **Transformation key**

  The versioned key that LDT applies to transform the data from the key used for initial data transformation. When the transformation key rotates, it transforms the data from a previous version of the transformation key to a new version.

Note: Transformation key and versioned key are used interchangeably throughout this document.

As soon as LDT applies the policy to a GuardPoint and enables protection for it, LDT triggers an initial transformation from the current key to the transformation key.

Note: With DSM 6.1, you may select current or transformation keys for CBC or CBC-CS1 encryption modes.

When the transformation key expires, it generates the next version of the versioned key with new cryptographic material. The DSM then pushes the policy to the hosts. The policy now contains the new version of the key. This initiates a rekey process on the GuardPoint to transform data to the new version of the transformation key specified in the policy.

Users and applications can continue accessing data without any interruption during initial encryption and subsequent key transformations.

In contrast, when you use a standard (non-LDT) policy, you achieve data transformation and protection with two separate policies. The first policy encrypts data with a non-versioned key. During this process, user and application access to data is denied. When the data transformation is completed, the second policy is applied to the GuardPoint. The second policy uses the same encryption key as the first policy, and it protects data and allows user and application access. The cryptographic material of the key does not change over the lifetime of the policy on that GuardPoint.

Note: During LDT policy creation, you must use the Apply Key effect in your policy. If you do not, then end users can see the clear text data until the file is transformed.

LDT Runtime Flow

This section presents an overview of how LDT works and what to expect when LDT is enabled and running in your environment. All of the tasks mentioned here are described in more detail later in this chapter.

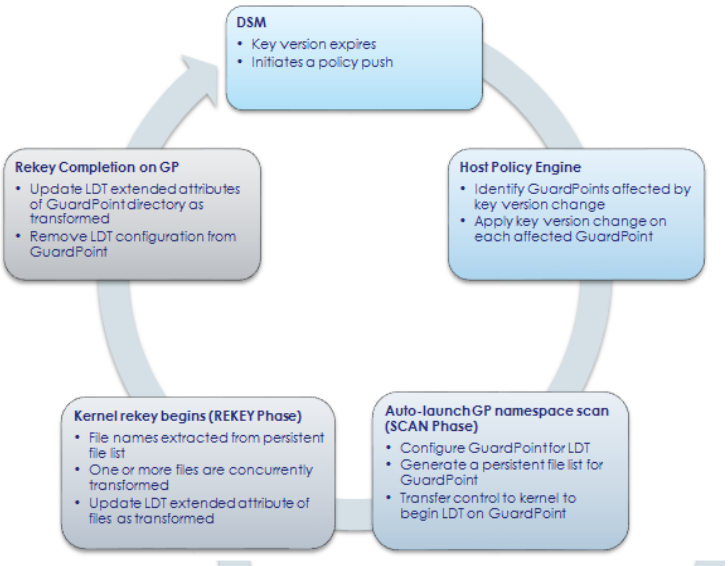
First, the administrator completes LDT setup:

1. Upload the LDT license on the DSM.
2. Register LDT hosts with the DSM and be sure that the hosts are licensed for LDT.
3. Create one or more versioned keys.

   For each key, specify the **Expiration Date** and the **Key Version Life Span**.
4. Define LDT policies which use the versioned key(s) and contain rules governing LDT operations.
5. Provide Quality of Service (QoS) settings for the LDT hosts. The QoS settings control the:
   - Impact of LDT on application workload by allowing the administrator to specify the time windows when LDT is permitted to operate.
   - Percentage of CPU resources that LDT can use, or the amount of data to transform according to the QoS setting per the DSM administrator.
   - Overhead of LDT I/O operations on the overall system performance using the voradmin command.

The previous step is optional, but highly recommended as a best practice. When these items are set up, LDT is ready to transform and encrypt data by applying policies to GuardPoints for live initial transformation and subsequent rekeys, as well as enforcement of security rules. The following figure shows the sequence of events during ongoing usage of LDT.

Figure 1-1: Phases of LDT runtime operation after keys and policies are defined



1. **Initial data transformation starts or key expires**

   Live Data Transformation begins when an LDT policy is first applied to a GuardPoint, or when a current key version expires. The DSM pushes the new policy, or the notification of a key version change, to the hosts that are protected by those policies.

2. **New key version triggers a rekey on the affected GuardPoints**

   On each host, VTE determines which GuardPoints are using the key that has just rotated to a new version. VTE starts an LDT rekey on each of those GuardPoints. If another rekey is already underway on that GuardPoint, the new rekey is queued for later execution. For details, see “Rotating Encryption Keys While a Rekey is in Progress (Relaunch)” on page 46.

3. **Scan for files**

   On each GuardPoint where VTE has started a rekey, LDT determines which files to transform. LDT takes inventory of files encrypted with earlier versions of the rotated key and makes a persistent list of the files for transformation. During this phase, the Rekey Status in the GuardPoint Status window of the DSM UI displays Starting, then Scanning.

   The scan phase might be interrupted, such as by a host reboot. In this case, when the host reboots and the GuardPoint is enabled again, the scan operation starts over from the beginning.

4. **Rekey/Key Rotation**

   a. Each file, from the persistent list of files, is decrypted using the old version of the key. The old key is applied to each file and then re-encrypted using the new version of the key. Note that new files created during the LDT process do not need to be rekeyed, as they inherit the new version of the key. Multiple files and multiple regions of files are rekeyed simultaneously.

   b. The LDT extended attribute of each file is updated. (For more about extended attributes, see “LDT Metadata in Extended Attributes” on page 59.)

   c. You can suspend and resume the LDT rekey operation manually, or through the QoS schedule. This manages the impact LDT has on other applications and processes.

      During this phase, the Rekey Status in the GuardPoint Status window of the DSM UI shows Rekeying or Suspended.

      If system errors occur during rekeying, such as IO errors or crashes, LDT can manage and recover from them after the system error is fixed.

5. **Finish**

   When all of the required files in the GuardPoint have been rekeyed, the system and storage resources used by LDT are released, except for the storage required for the extended attributes.

   LDT creates a rekey report, listing all of the files that were rekeyed. For more information, see “Obtaining a Rekey Report” on page 85.

   Upon completion of rekey, the Rekey Status in the GuardPoint Status window of the DSM UI shows Rekeyed.

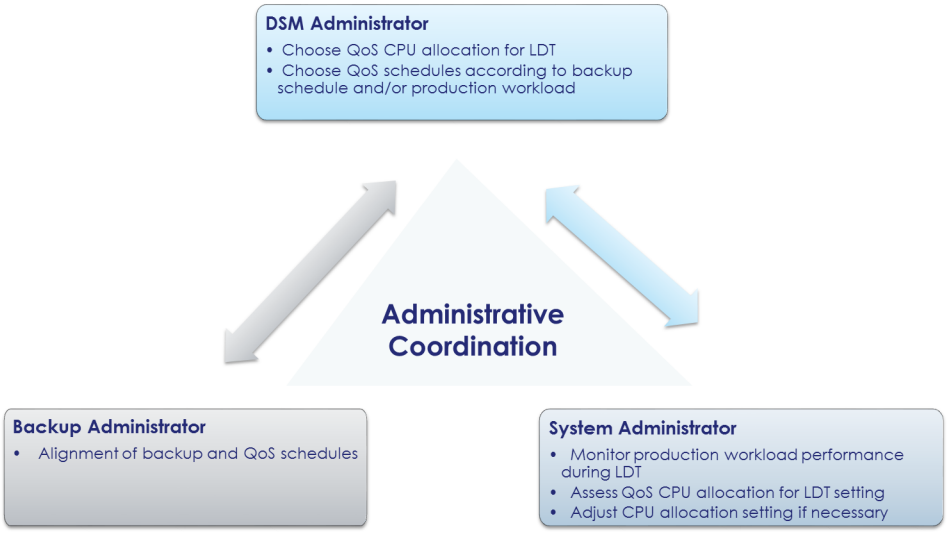
LDT Administrator Roles

Using LDT requires coordination and collaboration between several different administrators. Security, system, application performance and backup schedules are critical factors that affect planning. It is very important for the **DSM security administrator** to coordinate with the **system administrator** and the **backup administrators**. The following table describes the roles and responsibilities for these administrators.

Table 1-1: LDT Administrators

Role	Responsibility	Actions
DSM security administrator	Administers Vormetric Data Security Manager (DSM)	Coordinates with system and backup administrators. Creates security policies and rules for LDT. Creates Quality of Service schedule for LDT, applications, and backups.
System administrator	Administers servers on which Vormetric Transparent Encryption is deployed	Coordinates with the DSM security administrator to create Quality of Service schedule, taking into consideration the backup schedules.
Backup administrator	Manages data backups for data encrypted by Vormetric Transparent Encryption	Coordinates with security administrator to create Quality of Service schedule, taking into consideration the backup schedules.

The following process is recommended for coordination between the various administrators in order to achieve optimal QoS schedule settings and allocation of CPU resources when using Live Data Transformation.



Resiliency

Live Data Transformation is resilient to many user actions and system occurrences. Since it is designed to run periodically without intervention, it includes various features to provide this resilience.

Before LDT enables a GuardPoint, it checks for any inconsistencies in files that were undergoing rekey at the time when system operations were interrupted. If it finds any inconsistencies, LDT corrects them before it enables the GuardPoint. It should only take a few seconds to identify and correct any inconsistency. On Windows, this process is delayed until applications access the affected files.

During rekey, if an issue such as an I/O problem or system crash occurs, the resiliency features of LDT ensures the consistency of user data. Such issues can cause an interruption in the middle of a rekey operation. When system operations resume, LDT corrects the problems and then resumes rekeying.

Chapter 2: Getting Started

This chapter describes how to get started with LDT. It contains the following sections:

Using LDT

Note: If you are new to VTE and LDT, read the Vormetric DSM Administrators Guide first to familiarize yourself with the concepts of GuardPoints and Policies.

Note: Before installing LDT, refer to the compatibility matrix to verify that your version of Linux is supported. Contact Thales Technical Support (https://help.thalesesecurity.com/hc/en-us) for the matrix.

The following list contains the steps for successfully setting up and using LDT.

1. Install VTE Agent and LDT

   VTE Agent installs on your protected host. Live Data Transformation installs with the VTE installation.

   - See Chapter 2 in the *VTE Agent Installation and Configuration Guide* for a Linux installation.
   - See Chapter 3 in the *VTE Agent Installation and Configuration Guide* for a Windows installation.
2. Install the LDT License

   Live Data Transformation is a separately-licensed feature of VTE. Before you can use it, you have to install the license to activate it. LDT is licensed for a specific number of hosts.

   - See Chapter 2, "Licensing" in the *Vormetric Data Security Manager Installation and Configuration Guide*.
3. Register the LDT host with the DSM

   After you install the LDT license, the next step is to enable LDT on the hosts that contain the data you want to protect.

   - See "Registering an LDT Host" on page 22 for more information.
4. Create Versioned Keys

   LDT uses *versioned keys*. A versioned key rotates to the next version of the key generating new key material automatically without policy change. LDT encrypts data with keys that use encryption standards like AES-128 and AES-256. This allows data to be re-encrypted without users having to edit the policy.

   - See "Keys in LDT (Versioned Keys)" on page 12 for more information.
   - See "Creating and Viewing Versioned Keys" on page 23 for more information.
5. Create LDT Policies

   LDT uses a single policy to address both initial encryption and subsequent rekeying. The same policy applies to production access and security rules without restricting user or application access to data. Applications have continuity of access to GuardPoint data during LDT.

   - See "Creating LDT Policies" on page 25 for more information.

6. Set QoS Settings

   The purpose of QoS is to enable administrators to manage and control LDT impact to application workloads by monitoring and controlling the use of host system resources, such as memory or I/O utilization, during Live Data Transformation. Administrators can also choose schedules for data transformation, or manually pause or resume transformation operations.

   - See "Quality of Service" on page 27 for more information.

7. Create GuardPoints and Apply LDT policies to the GuardPoints

   A GuardPoint is a directory in the file system hierarchy, where its contents have a Vormetric data protection policy applied to it. The File System Agent intercepts any attempt to access anything in the GuardPoint and uses the policies obtained from the DSM to grant or deny the access attempt. Typically data copied into a GuardPoint is encrypted, and only authorized users can decrypt and use that GuardPoint data.

   - See "Create an LDT GuardPoint and Apply LDT Policies" on page 39 for more information.

Backup/Restore

Before you enable GP with LDT policies, make sure that you back up your data.

Note: While backing up, you must suspend LDT for the entire backup process.

- See "Backing Up and Restoring LDT GuardPoints" in Chapter 4, "LDT Administration" in the LDT Guide, for more information.

Restrictions

Remember the following restrictions when using LDT:

- LDT does not support nested GuardPoints, where a file system is mounted inside a GuardPoint namespace.
- For HA clusters, LDT only supports the Asymmetric (active/passive) configuration. LDT does not support Symmetric (active/active) configuration.

Windows Only Limitations

- LDT does not support GuardPoints on CIFS network shared directories.

Linux Only Limitations

- LDT does not support Linux automounted file systems.
- LDT does not support system hibernation (`pm-hibernate`) on Linux hosts where LDT is in use.
- LDT does not support GuardPoints on NFS shares and `secvm` configured devices.
- Do not use LDT on GuardPoints over XFS and VxFS file systems at the same time.
- You cannot use LDT and Docker container on the same host.
- You cannot use LDT and OpenShift container on the same host.

Chapter 3: Setting Up Live Data Transformation

This chapter describes how to prepare your system to use LDT. It contains the following sections:

System Requirements

Live Data Transformation (LDT) requires the following environment.

Note: Refer to the latest VDS/VTE Compatibility Matrix, in your online support portal, for a list of VTE versions and supported operating systems.

Windows System Requirements

This section describes the system requirements when using LDT with Windows-protected hosts.

Vormetric Software Requirements

- VTE Agent version 6.1.x or higher installed on the host
- DSM version 6.1.x or higher
- LDT license for each protected host

Windows Host System Requirements

- Host memory: Minimum of 8GB.
- Windows host disk space: LDT requires a specific amount of disk space in the file system, over and above the space required for the guarded files themselves. LDT uses the additional space to store LDT metadata. The typical minimum space requirement for a GuardPoint is the number of files in the GuardPoint multiplied by 4K, plus 256MB.
  - Use the following command to obtain accurate space requirements:

```
# voradmin ldt space <GuardPoint>
```

Backups

- The backup application must have the capability to back up user-extended attributes. The application must use Volume Shadow Copy service (VSS) for backup and restore. For example, Vormetric tested LDT with NetBackup. You can also use other applications that use VSS and can back up user extended attributes.

Linux System Requirements

This section describes the system requirements when using LDT with Linux protected hosts.

Vormetric Software Requirements

- VTE Agent version 6.0 or higher must be installed on the host.
- DSM version 6.0 or higher.
- LDT license for each protected host.

Linux Host System Requirements

- Host memory: Minimum of 8GB.
- Linux host disk space:

  LDT requires a certain amount of disk space in the file system, over and above the space required for the guarded files themselves. The additional space is used to store LDT metadata. To estimate the amount of free file system disk space required by LDT on a Linux host, use the `voradmin ldt space` command.

Network File Systems

To use LDT on an ext3 or ext4 file system, block size **must** be 4K. Run `dumpe2fs` to determine the block size of the ext3 or ext4 file systems before using LDT. This limitation does not apply to XFS or VxFS file systems. You can use LDT on VxFS and XFS with a block size of 1K or 2K.

For LDT to work properly, the underlying file system must support and enable user-extended attributes. All of the file systems supported by LDT support these attributes. If you are using LDT with ext3 or ext4 mount points, you must explicitly enable the extended attribute mount option by editing `/etc/fstab` and adding the `user_xattr` mount option. In the other file systems supported by LDT, user extended attributes are enabled by default, so you do not have to explicitly enable them.

Example `/etc/fstab` entry for ext3 on Red Hat 6 or SLES:

```
/dev/sdb1 /disk2 ext3 defaults,user_xattr 0 0
```

Example `/etc/fstab` entry for ext4 on Red Hat 6 or SLES 12 (ext4 is not supported on SLES 11):

```
/dev/sdc1 /disk3 ext4 auto,users,user_xattr,exec 0 0
```

For more information about extended attributes, see “LDT Metadata in Extended Attributes” on page 59.

Supported Clusters in Linux

Veritas Cluster Server in asymmetric (active/passive) mode, see the compatibility matrix.

Supported Applications in Linux

For all of the supported operating systems for database applications, see the compatibility matrix.

Backups

- The backup application must have the capability to back up user-extended attributes. For example, Vormetric tested LDT with NetBackup. You can also use other applications that can back up user-extended attributes.

Replication

- rsync
- Hardware/software based replication system

SAP HANA Fibre Channel Systems

SAP HANA is compatible with LDT. See the compatibility matrix.

Veritas InfoScale

For Oracle with Veritas Infoscale, Thales recommends that you set the following when you plan to use an LDT GuardPoint with an Oracle database:

- Set `filesystemio_options` to **SETALL**

Installing the Agent Host Software

Install the VTE 6.x agent and Live Data Transformation on the protected host. Follow the instructions in the *Vormetric Transparent Encryption Agent Installation, Configuration and User Guide*.

- See Chapter 2 in the *VTE Agent Installation and Configuration Guide* for a Linux installation.
- See Chapter 3 in the *VTE Agent Installation and Configuration Guide* for a Windows installation.

Setting the Linux Kernel Time Zone

The Linux kernel contains an internal time structure that may or may not contain time zone information. On system configurations that do not contain time zone information, LDT stores and displays timestamps for rekey beginning and ending in UTC (Coordinated Universal Time) rather than the system's local time zone. If this occurs, the administrator can set the kernel's internal time zone to the local time zone if they desire timestamps in their local time zone.

To set the Linux Kernel time zone information, at boot time type:

```
# hwclock --systz
```

The command sets the kernel's time zone to the local time zone and resets the System Time based on the current time zone.

Note: On systems that do not set the time zone by default, existing timestamps for completed rekeys remain in UTC, even if you run `hwclock --systz`. Only timestamps for new rekeys display the local time zone.

Installing the LDT License

Live Data Transformation installs with the VTE installation, but is a separately-licensed feature of VTE. Before you can use it, you have to install the license on the DSM to activate LDT on the VTE host. LDT is licensed for a specific number of hosts.

1. Obtain an LDT license from Vormetric. You can purchase LDT along with the DSM software, or you can add LDT later.
2. Install the license on the DSM. If you purchase LDT with the DSM, it includes the LDT license in the same license file with the DSM license, so there is only a single file to install.

If you add the LDT license later, you need to install an additional license file. For information about how to install the license, see the *Vormetric Data Security Manager Installation and Configuration Guide*.To confirm that the LDT license is in effect:

1. Ask the DSM Administrator to log on to the Management Console and click **System > License**.
2. In the Agent Type column, find the row labeled FS. The number of host licenses displays in the LDT License column of this row.

Registering an LDT Host

After you install the LDT license, the next step is to enable LDT on the hosts that contain the data you want to protect.

Note: The LDT license is valid for a certain number of hosts. Once you reach this limit, you can either purchase additional LDT host licenses, or reclaim a license by removing the LDT host. See “Removing LDT and Security Encryption” on page 78.

Add your host entry to the DSM:

1. On the Management Console, click **Hosts** > **Hosts**.
2. Click on the host name for an existing host or click **Add** to add a new host.
3. Add a Host Name.
4. In the password creation method, select **Automatically** or **Manually**.
5. In Registration Allowed Agents, select **FS**.
6. Select **Communication Enabled.**
7. Click **OK**.
8. On the DSM Management Console, select your protected host.
9. On the Edit Host page, select the **Live Data Transformation** option. Then click **OK** or **Apply** to enable LDT on your protected host.
10. Verify, on the DSM Management Console, that your host is enabled for LDT. Click **Hosts** > <*hostname*> and verify that the Live Data Transformation box is selected.

Chapter 4: Using Live Data Transformation

Setting up keys, policies, and GuardPoints with LDT is very similar to performing the same tasks in VTE without LDT. However, there are some differences.

This chapter contains the steps, in order, for successfully setting up and using LDT.

Creating and Viewing Versioned Keys

In LDT, you create a **versioned** key for an LDT policy and define the **life span** of the key. The life span is the duration (in days) of the versioned key. You also define an **initial expiration date**. When the key reaches its expiration date, it automatically rotates to a new version.

Although an LDT policy specifies the same key name, when the key rotates, the rotation of the key automatically starts live data transformation on the data in the GuardPoints protected with the LDT policy. There is no need to change the policy. However, you can manually rotate the key if circumstances require it.

Use the **Add Agent Key** page in the DSM Management Console when creating versioned keys.

CAUTION

Make sure that you are working in the Symmetric tab. LDT is not supported for asymmetric keys.

Creating a New Versioned Key for LDT

1. Refer to "Adding Agent Keys" in the *Vormetric Data Security Manager Administrators Guide*. This procedure explains all of the steps and settings, including those related to versioned keys.
2. Set the following required fields:
   - **Expiration Date**: LDT uses the initial transformation key specified in the LDT policy until this date occurs. On the expiration date, LDT creates a new version of the transformation key. Thereafter, it creates new versions based on the Key Version Life Span.
   - You must specify the Expiration Date. Without an Expiration Date, a versioned key is always viewed as the initial key version. This means that it does not trigger live data transformation during initial transformation or subsequent key rotations.
   - **Automatic Key Rotation**: When checked, this makes the key a versioned key. You can only use versioned keys for LDT.

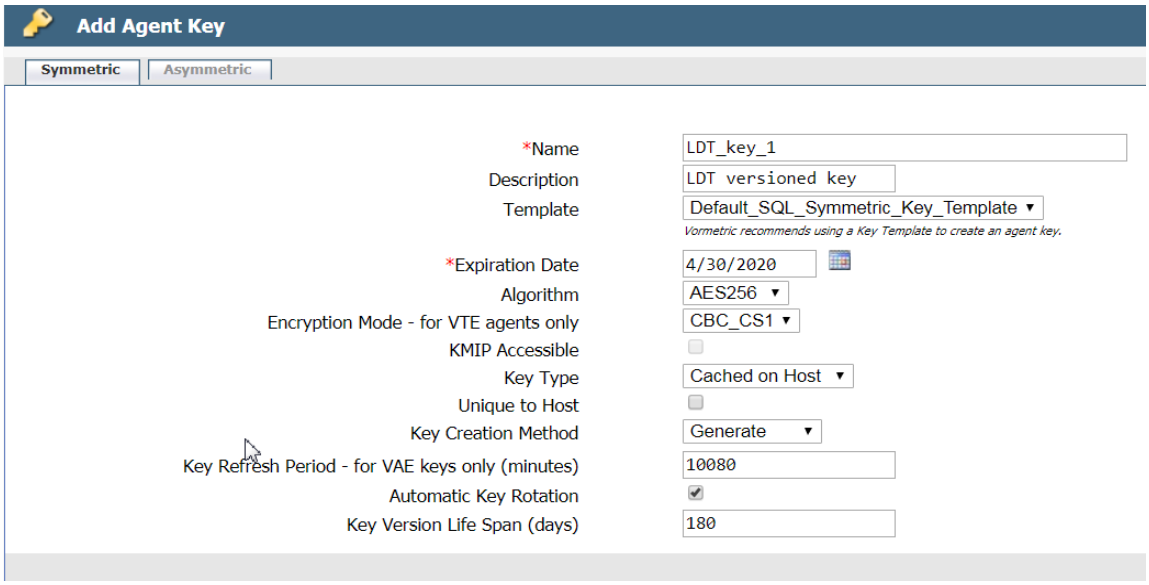
- **Key Version Life Span (Days).** Frequency of key rotation, in days, that determines how often the DSM automatically generates a new version of the key, after the initial expiration. This triggers Live Data Transformation on all GuardPoints using the key. Applies only if Automatic Key Rotation is selected.

Note: It is important to understand the interaction between the Expiration Date and the Key Version Life Span. LDT uses the initial version of the key until the Expiration Date. The Life Span has no effect until after the Expiration Date passes. Unless an administrator forces a rekey manually, there is no key version rotation and no Live Data Transformation until the Expiration Date arrives.

When creating or modifying a versioned key, set the key version life span with care. The rekey process uses system resources, and it may contend with host applications that also require these resources. The more often you rekey, the more often the LDT process requires system resources. Determining the life span is a balance between security, compliance, and convenience. If you have no contention problems between the LDT process and your host applications, you can rekey frequently. If you do encounter contention problems, then you may want to choose a longer interval between key versions.

The following figure shows an example of a versioned key. The key name is LDT_AES256_KEY, and it was created on January 1, 2016. It has a key version life span of 180 days. Its expiration date was January 1, 2017. The DSM generates a new version of this key on January 1, 2017, and then generates a new version every 180 days.

Figure 4-1: Creating a Versioned Key



For information about how versioned keys are rotated throughout the lifetime of the LDT policies that use them, see “LDT Runtime Flow” on page 14. For information about how to manually rotate a versioned key at any time, see “Rotating Encryption Keys (Rekey)” on page 41.

Obtaining Information About Keys

To obtain information about keys, use the DSM Management Console UI. For example, you might use this procedure to:

- Find out which keys are versioned keys, and therefore available for use in LDT policies.
- Locate a past key version so you can use it to decrypt old data.

To view keys in the DSM UI

1. In the Management Console, click **Keys** > **Agent Keys** > **Keys**.

   The currently defined keys display. The list can be long, so LDT provides a Search box so you can search for keys by name.

2. To find your versioned keys, look for a checked box in the Versioned column.

   Look at the Current Version column. If the current version is 0, this is the first version of the key. The key has not yet been rotated to a new version. If the current version is greater than 0, the key has been rotated. Either the key version expired, triggering an automatic rotation, or an administrator manually initiated a rekey, (see “Manual Key Rotation” on page 42).

3. To find the most recent version of a key, click the name of the key in the Name column. The General tab shows the most recent version of the key.

4. To find previous versions of a key, click the name of the key in the Name column.

   In the Edit Agent Key window, click the Versions tab. The key versions are listed in reverse chronological order (most recent first). If the key has been rotated many times, the list takes up multiple pages. Click the arrow buttons to move from page to page. On the last page, you can see the first version of the key, version 0.

Creating LDT Policies

LDT uses a single policy to address both data transformation and ongoing protection. In contrast, if you do not use LDT, you need a separate policy for initial data transformation, or rekey, and another policy to protect the data while it is in production use. For more information about LDT policies, see “Live Data Transformation Policies” on page 13.

To create an LDT policy:

1. Select **Policies** > **Manage Policies** > **Manage Policies**.
2. Click **Add**.
3. In Policy Type, choose **Live Data Transformation**.

   A default security access rule is automatically added in Security Rules. The action is `key_op`, and the effects are Permit and Apply Key. This rule permits key operations, for the policy, on all resources, without denying user or application access to resources. This allows it to perform a rekey operation whenever the encryption key rotates to a new version. This rule is required by LDT, so you cannot edit it, move it, or delete it.

4. Fill in the other fields as appropriate for an LDT policy. Some of the fields have special considerations when you use LDT:
   - **Policy Type**

     Must be Live Data Transformation, as noted earlier in these steps.
   - **Learn Mode**

     Learn Mode provides a temporary method for disabling the blocking behavior of VTE/LDT policies. While useful for quality assurance, troubleshooting, and mitigating deployment risk, Learn Mode is not intended to be enabled permanently for a policy in production. This prevents the policy Deny rules from functioning as designed in the policy rule set.

     Ensure that the policy is properly configured for use in Learn Mode. Any Security Rule that contains a Deny effect must have Apply Key applied as well. This is to prevent data from being written in mixed states, resulting in the loss of access or data corruption.

Note: Apply Key will have no effect when combined with a Deny rule unless the policy is in Learn Mode.

- **Security Rules**

  The Security Rules panel contains the policy rules for the production workload. One default rule is added automatically to every LDT policy, as described in step Step 3. After this default rule, the other security rules are specified in the same way, whether or not LDT is in use.

  If you set the "**When**" clause of key_op rules in LDT and a time period is set, it affects user I/O operations, not LDT I/O operations. LDT starts rekeying the GuardPoint regardless of the value in the key_op rule's When clause. LDT performs rekey operations according to the QoS schedule assigned to the host.

- **Key Selection Rules, Current Key**

  Indicates the state of the data before the LDT policy is applied to the GuardPoint. The data might be clear text, or it might already have been encrypted using a non-versioned key.

  Specify *clear_key* in Current Key if the data has not yet been encrypted. Otherwise, specify the name of the non-versioned key that was used to encrypt the data into its current state.

- **Key Selection Rules, Transformation Key**

  The **Transformation Key** is the versioned key that is applied to the data for its initial transformation. After the initial transformation, it rotates every time its life span ends. When the key rotates, all of its properties, including the key name and cryptographic algorithm, remain unchanged, except for the key material for encryption. Specify the versioned key name in the **Transformation Key** column. You can enter `clear_key` to transform your data to clear using LDT. `clear_key` is an exception to a non-versioned key in Transformation Key.

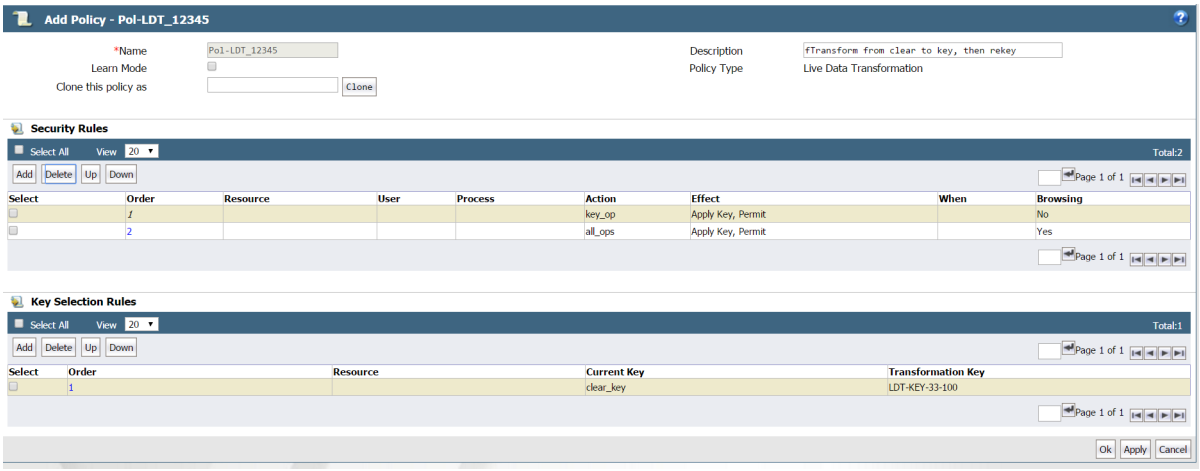
- **Key Selection Rules,** Resource

  Allows you to add or edit a resource set. A resource set is a named collection of directories, files, or both, to which a user or process is permitted or denied access.

Note: Thales strongly recommends limiting the number of resource sets in LDT policies to 50 resources or less.

The following example shows a simple LDT policy that encrypts clear-text files using a versioned key named LDT-Key-33-100, as shown in the Key Selection Rules panel. The example `key_op` and `all_ops` actions in the Security Rules panel grant user and application access to files at all times, including during initial LDT and subsequent key rotations.

Figure 4-2: LDT Policy that encrypts clear-text files



Quality of Service

Live Data Transformation runs in real time, while users actively interact with applications. This could impact performance. However, LDT is designed to not adversely affect application or system performance.

Purpose of QoS

Quality of Service (QoS) provides tools for an administrator to minimize the effect of LDT on system and application performance. It provides a set of parameters that administrators can set to control LDT use of system resources, primarily CPU and IO bandwidth. When the QoS parameters are set appropriately, LDT stays within the defined boundaries to ensure that critical user applications are not adversely affected by LDT operations.

Manage LDT impact

Administrators can pause or resume LDT operations to manage and control LDT impact to application workload. When data transformation occurs, either during initial or subsequent transformations, it requires substantial host CPU and I/O resources. This can cause contention for resources between the applications simultaneously running on the protected host. The administrator specifies QoS settings on each host, or at a host group level, that is using LDT. When Live Data Transformation is running, QoS monitors CPU or rekey/scan rate on the host and enforces the QoS settings. QoS can also monitor and enforce an administrator imposed limit on the volume of data undergoing rekey per second. The QoS settings enable you to strike a balance between completing an LDT process and not interfering with host application performance.

Monitor and control CPU usage

QoS monitors and controls the use of host system resources during Live Data Transformation, specifically, CPU usage and rekey/scan rate.

Note: You can control CPU usage or rekey/scan I/O rate, but not both. The CPU usage and rekey/scan I/O rate options are mutually exclusive.

Monitor and control rekey/scan I/O Rate

As of VTE Agent v6.1.2, you can choose Rekey IO Rate as a threshold to control the LDT processing rate. When this threshold is entered, the Quality of Service continuously monitors LDT transformation and enforces the specified amount of data during:

- **Rekeying**—LDT is transforming the data on active GuardPoints based on the new key version.
- **Scanning**—LDT is analyzing files in GuardPoints. Scanning occurs:
  - Before initial transformation (Linux only)
  - Before a rekey (Linux only)
  - Following an interrupted rekey, such as a reboot on Linux or Windows, and also a directory rename or directory deletion on Windows

The Rekey I/O Rate is either 0, or greater than 0. By specifying 0, Quality of Service resets the previously specified Rekey I/O Rate, if any, and then stops monitoring the LDT rekey/scan processing rate.

When the Rekey I/O Rate is set, Quality of Service ignores the CPU and/or the iow (IOWAIT) thresholds previously set. After resetting the Rekey I/O Rate to 0, the Quality of Service will resume enforcing the CPU and/or the iow thresholds, provided one or both are set.

To rekey at full throttle, you can set the Rekey I/O Rate, the CPU, and IOWAIT thresholds to 0.

Note: IOWAIT has been deprecated on Linux, and it is not supported on Windows. Instead, use Rekey IO Rate to control the LDT I/O utilization on both Linux and Windows.

You can enter a threshold for Rekey I/O Rate for one or more managed hosts on the QoS settings area of the **GuardPoint** tab under host configuration in the DSM (DSM version 6.2.0 or later) or on a single host using the `voradmin ldt ior <iorate>` command.

Note: If you set the Rekey I/O rate threshold using `voradmin` and that host is part of a DSM policy in which the Rekey I/O Rate was set in the DSM, the `voradamin` setting overrides the Rekey I/O Rate set in the DSM.

A tolerance level is associated with the Rekey I/O Rate. Together, the tolerance and Rekey I/O Rate specify a range for the LDT processing rate. The Quality of Service selects a proper tolerance for a Rekey I/O Rate provided through the `voradmin` command, and maintains the LDT processing rate at the specified Rekey I/O Rate plus or minus the tolerance. The tolerance is selected as follows:

- When the Rekey I/O Rate is less than or equal to 10MB/sec, the tolerance is 3MB/sec.
- When the Rekey I/O Rate is greater than 10MB/sec. and less than 50MB/sec, the tolerance is 4MB/sec.
- When the Rekey I/O Rate is at 50MB/sec or higher, the tolerance is 10% of the specified Rekey I/O Rate.

To set or reset Rekey I/O Rate on a single host, use the `voradmin` command as follows:

To set the threshold of 50 MB/sec., use the following command:

```
# voradmin ldt ior 50
```

To reset the current threshold:

```
# voradmin ldt ior 0
```

For more information about setting the Rekey I/O Rate using `voradmin`, see "Select and Set Rekey I/O Rate" on page 37.

You can also set the Rekey I/O Rate for one or more managed hosts on the QoS settings area of the **GuardPoint** tab under host configuration in the DSM (DSM version 6.2.0 or later). For more information about using the DSM method, see "How to Set QoS" on page 30.

Monitor and control I/O wait time (Linux only)

You can choose to add **I/O wait time** for monitoring and control. The I/O wait time threshold refers to the system's peak I/O wait time during its production workload. For a given threshold, you can specify a tolerance level that is the amount of I/O overhead allowed by LDT for rekeying data. By specifying the I/O wait time, and the tolerance level, QoS continuously monitors your system's average I/O wait time during LDT operations. It applies the tolerance factor to the average I/O wait time to adjust LDT I/O operations. This assures that the system's average I/O wait time remains within the desired range and does not exceed the threshold.

The desired range is the specified average I/O wait time (milliseconds) plus or minus the specified tolerance. **Tolerance** provides the range in which the average I/O wait time, which includes LDT I/O operations, is maintained.

For example, you may specify a desired I/O wait time during rekey. QoS maintains the desired I/O load for rekey according to the specified I/O wait time. If the total I/O wait time increases beyond the desired I/O wait time, QOS reduces the rekey I/O load by some percentage until the desired I/O wait time is achieved or the rekey I/O load reaches down to a preset N number of rekey operations.

CAUTION

The Linux I/O wait option has been deprecated in favor of Rekey I/O Rate. Support for the I/O wait option will eventually be removed from VTE. If you are currently using I/O wait time option with QoS, please plan on switching from I/O Wait Time to Rekey I/O rate.

QoS scheduling during Backup/Restore

QoS scheduling plays an important role when backing up/restoring data without the *Apply Key* rule applied to the backup/restore process. During backup/restore, you **must** pause LDT operations before taking backups. QoS scheduling allows the administrator to enter the schedule for QoS aligned with the backup schedule, and pause the LDT processes for the duration of the backup. The schedule specifies which days of the week, and what times of day, LDT is permitted to run. LDT cannot run at any time that is not permitted by the QoS schedule. QoS suspends LDT operations at all times outside of the schedule.

When setting a QoS schedule, consider your system and application peak demand periods during the day and week. Also consider your schedule for data backups. Schedule LDT to pause when you need all available system resources for other tasks, such as meeting peak user demand or performing data backups.

Note: On Windows, if your backup applications are using VSS, then you do not need to pause LDT on Windows.

How to Set QoS

Note: If you choose Rekey IO Rate threshold to control the LDT processing rate, you can skip the steps listed below for setting the CPU threshold for LDT.

To set QoS for a host to enforce a threshold on LDT based on CPU percentage:

1. Click **Hosts > Hosts >** <*hostname*> > **GuardPoint**.
2. Set the **Schedule** parameter:

   Choose one of the options from the dropdown list. By default, the list contains WEEKNIGHTS (rekey performed between 12 AM and 7:00 AM), WEEKENDS, or ANY_TIME. The default setting is ANY_TIME. By limiting LDT to periods of low application usage, you minimize the potential for resource contention between applications and LDT.

   You can add to the choices in the Schedule dropdown by creating custom schedules. See "Creating a Custom QoS Schedule" on page 31.
3. Enter the Rekey IO Rate threshold or percentage of CPU resources to allocate to LDT in the **Set % of available CPU usage for rekey** field:

   Choose a Rekey IO Rate threshold or the percentage of total CPU resources that LDT processes can use. LDT applies a tolerance of +/- 2% on CPU threshold settings up to 7%. For threshold over 7%, LDT applies a tolerance of +/- 4%.

   If CPU% is set to 0, QoS stops monitoring CPU usage and LDT operations run at maximum rekey rate within the available system resources. Setting CPU percentage to 0 does not affect LDT schedules, so LDT operations are suspended and resumed per QoS schedule.

   If you do not enter CPU threshold percentage, LDT applies CPU threshold of 5% capped by default.

   If CPU% is set to a very high value, such as 25%, the rekey process competes with other applications to use as many CPU cycles as it can. As a best practice, start with a setting of 10%, and increase or decrease it slowly by 5% until it reaches a reasonable level that does not adversely affect the performance of user applications. The higher the percentage, the more quickly LDT completes its processing. However, this speed causes increased competition for resources, which can significantly degrade the performance of other applications using this host.

CAUTION
Do not set CPU% to a very high value, or to 100%, in an attempt to force faster data transformation. This can potentially exhaust other system resources.

4. Set the **Cap CPU Allocation** for rekey.

   Check this box to specify that the CPU allowance must never exceed the percentage set in CPU%. If Cap CPU Allocation is not checked, and additional CPU resources are available on the host, LDT consumes part of the available resources for rekey above the CPU threshold. Exceeding the threshold may impact your production workload as your production CPU resource consumption fluctuates over time.

   For example, if CPU% is set to 10%, but Cap CPU Allocation is not set, the rekey process continues consuming available CPU cycles after reaching 10% CPU utilization, at which point the rekey process starts contending with applications for CPU cycles.

Creating a Custom QoS Schedule

LDT provides three predefined QoS schedules: ANY_TIME, WEEKENDS, and WEEKNIGHTS. You can define additional custom schedules.

1. In the DSM, choose **Hosts > QoS Schedules**.
2. Click **Add**.
3. Enter a unique name for the new schedule.

   This name displays in the **Schedule** dropdown list when you set QoS parameters.
4. Click **Add**.
5. Choose the start day/time and the end day/time for LDT operations.
6. Click **OK**.

QoS Best Practices

This section gives tips and examples to help you set QoS parameters for the best results.

General Best Practices for QoS

- Use Rekey I/O Rate threshold to limit LDT impact, if any, to your production workloads. Rekey I/O Rate approach is a simpler method for DSM or system administrators to enforce a limit on the volume of data that LDT should rekey per second. You can choose a threshold, in units of MBs per second, which is a small percentage of peak IOPS from your production workload.
- You will see the effects of QoS settings only if the number and/or types of files in the GuardPoints stress the rekey or scan processes. On hosts with a relatively small number of files, the rekey or scan process may complete quickly without hitting a threshold and causing throttling to occur.
- Use QoS CPU parameters as an alternate method for controlling the effect LDT has on application performance

  Set limits on LDT CPU usage whenever runtime monitoring shows that user applications are affected by LDT. Start by setting the CPU parameter to 10%, then increase or decrease in 5% intervals, as needed, to tune the CPU allocation. When an acceptable level is reached, and LDT is not noticeably affecting user applications, leave the QoS CPU parameters at a constant setting.
- Use monitoring tools

  Monitor host CPU utilization with tools like `vmstat`, `top`, and `iotop` on Linux and `perfmon` on Windows.

  You can also monitor and obtain statistics with `voradmin`:

  ```
  voradmin ldt stats
  ```

  For more information about `voradmin ldt stats`, see “Obtaining LDT Statistics at the Command Line” on page 85.
- **Select CPU resource allocation for LDT from 1% to the available limit minus 5%**

  If the monitoring tools indicate system CPU usage, without LDT, it is at N%, available CPU resource is M%, where M = 100 - N. Select a percentage within 1 - (M - 5) to allocate to LDT CPU usage. However, remember that QoS tolerates 2% - 4% leeway in the actual CPU usage, so adjust your selection by 2 - 4%.
- Do not set CPU resources to 0% or 100% in an attempt to minimize or stop LDT

  A CPU% value of 0 or 100 is reserved for disabling the QoS CPU monitoring function. This does not stop LDT or minimize its resource usage; rather the opposite. It enables LDT to run with its maximum rekey rate. Note that when CPU % is not set, and it shows 100% setting on the DSM, LDT hosts installed with VTE v6.1 enforce a 5% CPU threshold by default. We recommend that you change the 100% setting to 5% capped on the hosts installed with VTE versions lower than v6.1.

- Cap the CPU allocation

  QoS provides a CAP CPU Allocation parameter. Set this parameter to True. This ensures that LDT resource usage never exceeds the allocated percentage.
- Apply **iowait** threshold very carefully to try controlling LDT I/O load

Example: Setting QoS before starting LDT

You can be proactive and set up QoS parameters before enabling GuardPoints that are protected with LDT policies. This ensures QoS starts monitoring and controlling LDT resource usage from the start. The following graph shows an example where 10% of the CPU is assigned to LDT. QoS makes sure that LDT is restricted to use only 10% of the CPU. There is a tolerance level of +/- 4%, so actual LDT usage can range between 5% and 15% of CPU. In the following example, applications use 75% of the CPU resources. As the graph shows, when LDT starts, application CPU utilization drops for a moment, because LDT exceeds the CPU threshold. QoS immediately reduces LDT's CPU usage to 12%, which is within tolerance levels for a 10% setting, and the application CPU share returns to normal.

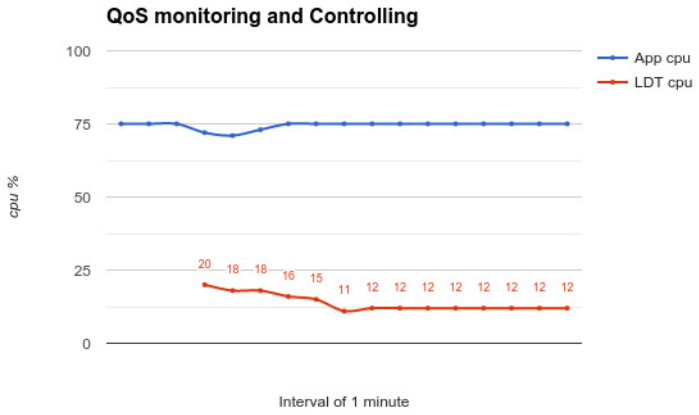


Figure 4-3: QoS makes visible improvement immediately when LDT starts

The graph above was obtained on a Linux system running sysbench.

- To find the amount of CPU resources currently in use by applications, type:

```
# top -n 1 -b | grep sysbench | awk 'BEGIN {cpu=0} {cpu += $9} END {print
cpu}'
```

- To find the amount of CPU currently in use by the LDT-protected host, type:

```
# top | grep Cpu
```

- To find the amount of CPU currently in use by LDT, type:

```
# voradmin ldt stats | grep CPU
```

Example: Monitoring and controlling resource usage during LDT

Suppose a Live Data Transformation has started with CPU set to 25%, and users realize their applications are affected. For example, there might be a higher than expected level of LDT I/O operations. To return application performance to normal, reduce the CPU allocation for LDT.

1. Set the **CPU parameter** to a lower value, such as 10%.
2. Select the **Cap CPU Allocation** option.
3. Set the I/O wait parameter.

QoS restricts LDT CPU usage to 10%. The application user should monitor their application. If the application's performance is still affected, reduce the CPU parameter further, such as to 5%. Repeat this procedure until application performance returns to a satisfactory level.

The following graph shows an example where QoS is not enabled to monitor and control LDT CPU usage from the start. When LDT starts, application CPU usage drops from 65% to 20%. By setting the QoS CPU parameter to 10%, application usage is greatly improved.

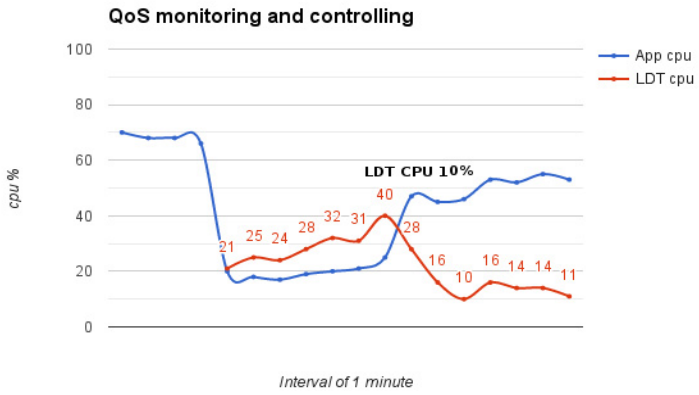


Figure 4-4: CPU usage allocation before and after QoS CPU parameter is set

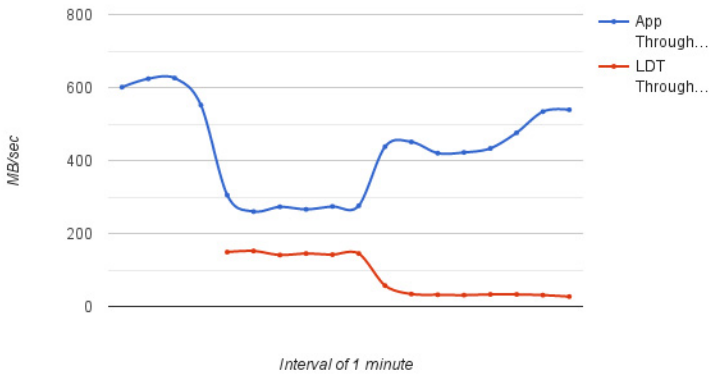
Example: How QoS CPU settings affect I/O bandwidth

Controlling CPU utilization indirectly controls I/O bandwidth. When LDT is consuming less CPU resources, it is usually performing fewer operations of all kinds, including input and output.

The following graph (Figure 4-5) shows how an application's I/O throughput is affected by LDT, and how QoS can reduce this effect through monitoring and controlling the LDT CPU resource usage.

This graph is from the same system described in "Example: Monitoring and controlling resource usage during LDT" on page 32. No QoS parameters are set at first. When LDT starts, application I/O drops from more than 600 MB per second to below 30 MB per second. By setting the QoS CPU parameter to 10%, the LDT I/O operations drop to below the application's ability to perform so application I/O operations is greatly improved.

Figure 4-5: I/O operations before and after QoS CPU limit is set



To obtain data for this graph:

- Use `iotop` and benchmarking tools on an RHEL system to obtain application I/O throughput
- To obtain the LDT I/O throughput and rekey rate, type:

```
# voradmin ldt stats
```

Determine and Set the I/O wait time

Monitor the average I/O wait time using `iostat` on Linux, or `perfmon` on Windows, during the production workload. Thales recommends monitoring the average I/O wait time during the peak levels of the workload.

1. Run the `iostat` command and specify the number and duration of intervals to determine `iowait` time.

   Example:

   ```
   # iostat -txm 10 2
   ```

Note: For Windows, use `perfmon` to obtain:
`\PhysicalDisk(_total)\Avg. Disk sec/Transfer`

2. Get the total I/O wait time from the sampled data. It is the sum of the await field of disks.

Note: Do not include await values from volumes and device mappers.

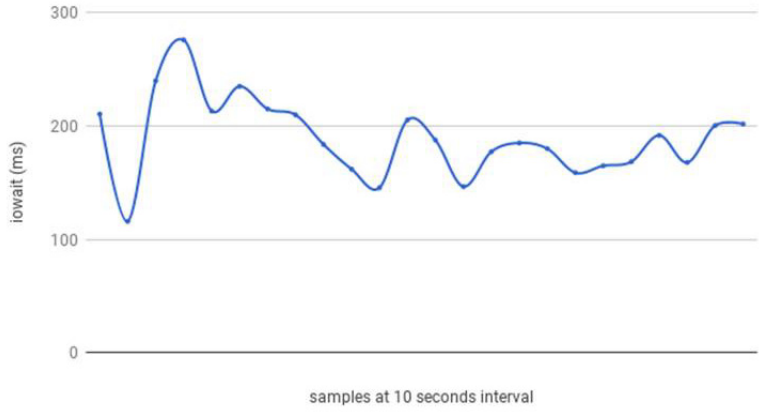
Example of iostat data

```
Monday 04 September 2017 03:13:59  PDT
avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           0.18    0.00    2.52   42.03    0.00   55.27

Device:         rrqm/s   wrqm/s     r/s     w/s    rMB/s    wMB/s avgrq-sz avgqu-sz   await  svctm  %util
fd0               0.00     0.00    0.00    0.00     0.00     0.00     0.00     0.00    0.00   0.00   0.00
sda               0.00     0.00    0.03    2.17     0.00     0.03    28.77     0.00    0.61   0.18   0.04
sdb               0.00     0.00    0.00    0.00     0.00     0.00     0.00     0.00    0.00   0.00   0.00
sdc               0.00     0.33    0.00    0.47     0.00     0.00    14.86     0.00    0.00   0.00   0.00
sdd               0.07    17.20    0.53   12.53     0.01     0.18    29.31     1.14   88.90  76.30  99.70
sde               0.00    75.57    0.07  839.20     0.00     4.04     9.86   141.87  169.15   1.19  99.94
sdg               0.00     0.00    0.00    0.00     0.00     0.00     0.00     0.00    0.00   0.00   0.00
sdh               0.00     0.00    0.00    0.00     0.00     0.00     0.00     0.00    0.00   0.00   0.00
sdf               0.00     0.00    0.00    0.00     0.00     0.00     0.00     0.00    0.00   0.00   0.00
dm-0              0.00     0.00    0.11    2.17     0.00     0.01     9.19     0.02    7.49   0.28   0.06
dm-1              0.00     0.00    0.00    0.00     0.00     0.00     8.00     0.00    1.91   1.14   0.00
dm-2              0.00     0.00    0.15    4.97     0.01     0.02    10.73     0.05    8.80   0.04   0.02

~
```

For the above example:

```
Total I/O wait = .61+ 88.9 + 169.15 = 258.66
```

3. Obtain multiple samples at 10 second intervals. Estimate the `iowait` time when it stabilizes.
   In the following chart, the I/O wait time is nearly 200ms when in a stable state.



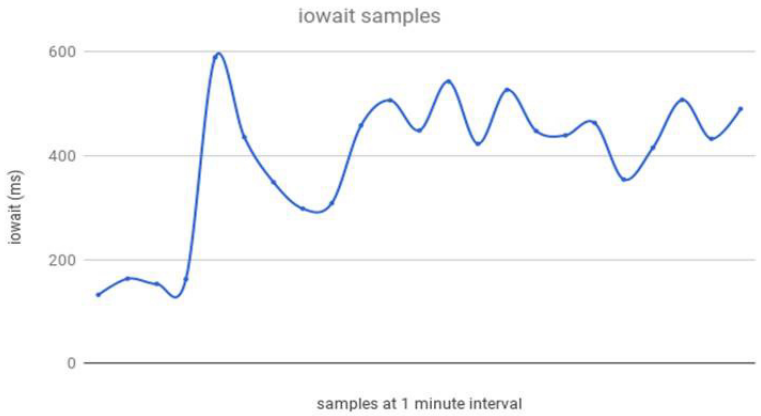
4. Estimate the `iowait` threshold by running LDT and watching the degradation in I/O wait time. Set the `iowait` threshold and/or tolerance parameters in units of milliseconds using voradmin command, type:

   `# voradmin ldt iow <iowait> [ <tolerance> ]`

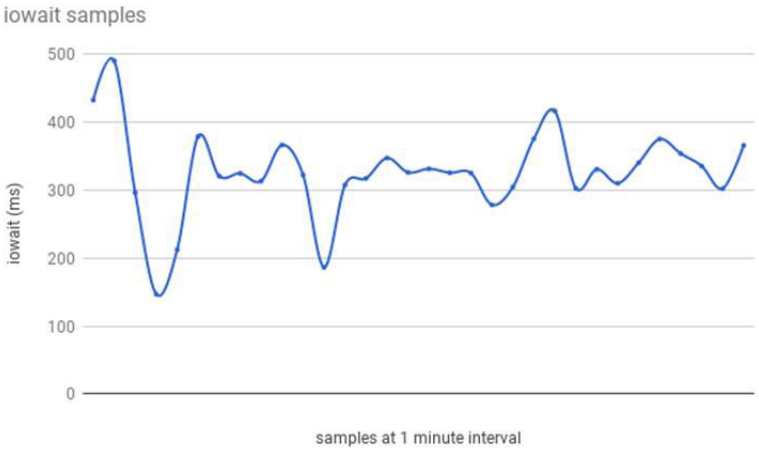
   Example:

   `# voradmin ldt iow 350 10`

   In the following graph, observe that once LDT starts, the total I/O wait time jumps to 600ms and oscillates between 400 to 500 ms. Without LDT, I/O wait time was 200 ms. If the application I/O wait time suffers because of LDT, reduce the LDT I/O usage by setting `iowait` threshold to a suitable level between 200 to 500 ms. In this example, we have chosen 350ms as the desired/suitable level of total I/O while LDT and application workloads are running concurrently.



From the graph, observe that QoS tried to control I/O wait time and now it oscillates near about 350 ms.



The table below explains what is monitored and controlled by QoS when CPU or I/O resource thresholds are set.

Resource Threshold Set	Monitored and Controlled by QoS
CPU (value must be greater than 0.)	CPU utilization to adjust LDT operations to maintain the set CPU threshold
I/O wait (value must be greater than 0.)	I/O average wait time to adjust LDT operations to maintain the set I/O wait threshold
CPU and I/O wait	CPU and I/O, both resources are controlled by their threshold limits
CPU and I/O wait set to 0	No resource is controlled or monitored

Select and Set Rekey I/O Rate

You can choose to set the Rekey I/O Rate to control I/O operations from LDT to minimize LDT impact to your production workload. It's assumed that you already know the maximum IOPS on your host system during your production workload. With this information, you can choose a threshold for Rekey I/O Rate and enforce the selected threshold during live data transformation. The work flow is as follows:

1. Set **Rekey I/O Rate** threshold using `voradmin` or in the DSM.
2. QoS retrieves the threshold and starts monitoring and controlling LDT according to the specified threshold and the tolerance factor corresponding to the threshold.
3. The selected threshold will be in effect within 2 to 4 minutes after entering the threshold.

When Rekey I/O Rate and CPU or IOWAIT thresholds are set, QoS will monitor and control the LDT processing rate based on the Rekey I/O Rate threshold. The CPU and/or IOWAIT thresholds will be ignored.

Set Rekey I/O Rate Threshold

Set **Rekey I/O Rate** threshold by using `voradmin`:

```
# voradmin ldt ior 10
```

You can also set the Rekey I/O Rate for one or more managed hosts on the QoS settings area of the **GuardPoint** tab under host configuration in the DSM (DSM version 6.2.0 or later). For more information about using the DSM method, see "How to Set QoS" on page 30.

In the `voradmin` example above, QoS enforces the threshold of 10MB/sec with the tolerance of +/- 3MB/second. Effectively, LDT attempts to rekey the amount of data in the range of 7MBs/second to 13MB/second.

On Linux and Windows, you can use `voradmin ldt ior` to report the current threshold setting without specifying a value for threshold:

```
# voradmin ldt ior
```

QoS Rekey I/O rate threshold: 10 MB/sec

QoS Rekey I/O tolerance: 3 MB/sec

Be sure the threshold you enter is appropriate for your production workload.

a. Observe the Rekey I/O Rate for a few minutes using `voradmin`.

On Linux, you can do this using:

```
# voradmin ldt stats

 Host level statistics:
 File stats: rekeyed=202390, passed=0, created=0, removed=0
 Data stats: rekeyed=6.2GB, truncated=0.0MB
 QoS: IOR threshold=10MB/sec, tolerance=3MB/sec
 current_rekey_rate=2MB/sec, current_iow=0ms
 load_factor=50, delay_factor=0, delay_scan=0
```

On Windows, you can do this using:

```
voradmin ldt monitor
Host Stats:
Total number of Guard Points    = 1
Rekey Status                    = Rekey done (Finished rekey on 1 out of 1
```

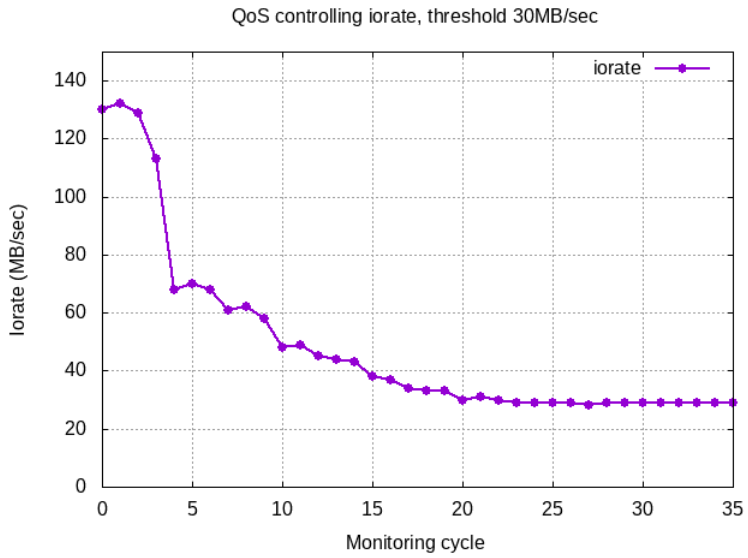
```
GP's)
Total files to be transformed  = 0
Total files transformed        = 0
Total files in progress        = 0
Total transformation threads   = 0

Current rekey rate             = 0 KB/s
Rekey IO rate threshold        = 1000 MB/s
Rekey IO rate tolerance        = 4 MB/s
```

b. Set an appropriate threshold. Do not set the threshold value too high, as QoS might not be able to achieve it because of other resource bottlenecks.

4. Check the QoS controlling rekey rate.

   QoS will monitor and control LDT utilization using the specified threshold. The following figure shows an example of how QoS monitors and controls LDT utilization. In this example, the threshold is 30 MB/sec. Throughput of LDT was nearly 130 MB/sec. QoS brings it down to within the range of 30 MB/second.

Figure 4-6: QoS controlling I/O rate, threshold 30MB/second



5. Disable QoS

QoS will not monitor and control resources when all the thresholds, CPU, Rekey I/O rate, and IOWAIT are set to 0. When Rekey I/O Rate and IOWAIT are not explicitly set, it is considered to be set to 90 MB/second.

QoS continues to apply its schedules for suspending LDT operations at certain days and times regardless of what values are set for CPU, Rekey I/O Rate, and IOWAIT thresholds.

Summary of QoS Resource

The table below summarizes the available thresholds and the actions of QoS module to enforce the set thresholds:

Scenario	QoS Action
Only Rekey I/O Rate threshold is set	Monitor and control the LDT processing rate based on Rekey I/O Rate
Rekey I/O Rate and CPU threshold are set	Monitor and control the LDT processing rate based on Rekey I/O Rate. CPU threshold is ignored.
Rekey I/O Rate, CPU, and IOWAIT thresholds are set	Monitor and control the LDT processing rate based on Rekey I/O Rate. CPU and IOWAIT thresholds are ignored.

Create an LDT GuardPoint and Apply LDT Policies

After you have installed the license and registered the LDT host (see "Setting Up Live Data Transformation" on page 19), you can create an LDT GuardPoint on the host. When you create the LDT GuardPoint, you also select and apply the policy and transformation keys to that GuardPoint.

This section describes two scenarios:

- Create an LDT GuardPoint on a previously unprotected directory.
- Convert a non-LDT GuardPoint to an LDT GuardPoint.

Create an LDT GuardPoint for an Unguarded Directory

To create an LDT GuardPoint on what was previously an unprotected/unguarded directory:

1. Create an LDT policy that transforms data from clear text to a versioned key. In the policy, set Current Key to `clear_key` and Transformation Key to the versioned key. See "Creating LDT Policies" on page 25.
2. Set, or modify, the Quality of Service (QoS) parameters to account for LDT on all GuardPoints. See "Quality of Service" on page 27.
3. Click **Guard**.
4. The Guard File System window displays. Set the following values:
   a. Select an LDT Policy.
   b. Select **Directory (Auto Guard or Manual Guard)**.
   c. Enter or browse for the directory to protect.
   d. Do **not** check Automount. LDT is not supported on automounted file systems.
   e. Check **Transform Sparse Regions** if you want those regions allocated with disk blocks.

      A *sparse region* is a region within the file size that has not yet been written to. Therefore, it is not allocated with disk blocks. Any attempt to read a sparse region reads stream of zeros as data. A file may have one or more sparse regions, or an entire file may be sparse.

      By selecting Transform Sparse Regions, LDT transforms a file without checking or skipping sparse regions, if they exist. Consequently, as LDT operations transform and fill sparse regions with encrypted stream of zeros, sparse regions are allocated with disk blocks. This increases the number of disk blocks utilized in the file system.

      By not selecting Transform Sparse Regions, LDT detects and skips transforming sparse regions. Therefore, it does not change the number of blocks utilized in the file system.

5. Click **OK**.

   The **GuardPoint** tab displays again. Live Data Transformation starts immediately after you enable the GuardPoint on the host.
6. Wait a moment and check the status of the GuardPoint in the Rekey Status column of the GuardPoint tab.
7. In the Status column, click the status icon (the green circle) to get status details about this GuardPoint.

   The GuardPoint Status window displays. For an explanation of all the items in the GuardPoint Status dialog, see “Obtaining Statistics in the UI with GuardPoint Status” on page 83.

Note: When you first apply a GuardPoint, the key version shown in GuardPoint Status is 0. This indicates the first version of the key.

8. Wait for LDT to complete the rekey process for the GuardPoint. When the Rekey Status column shows 100%, the GuardPoint has been rekeyed and the GuardPoint status changes to rekeyed. For information about the values in the Rekey Status column, see “Checking the Rekey Status” on page 42.

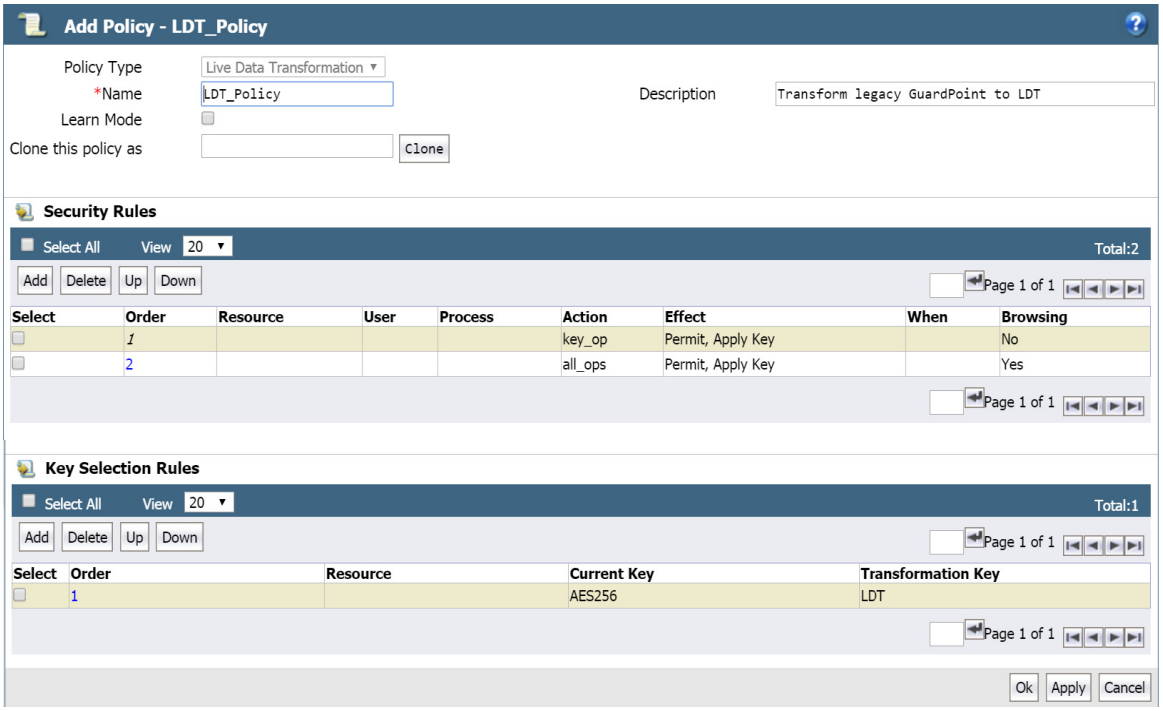
Converting a Non-LDT GuardPoint to an LDT GuardPoint

If you have an earlier version of Vormetric Transparent Encryption installed, you most likely have set up GuardPoints. These legacy GuardPoints are protected with policies that do not make use of Live Data Transformation. This means that you must stop all users from accessing the files in these GuardPoints while the policy performs encryption.

After installing VTE 6.0.x or higher, and setting up LDT on the hosts, you can change legacy GuardPoints to LDT GuardPoints. LDT GuardPoints provide the advantage of allowing users to access all files in the GuardPoints while encryption is occurring. There is no downtime for the user except for the time needed to apply the GuardPoint.

1. Write a new LDT policy that transforms data from the non-LDT/non-versioned key used in the existing GuardPoint, to an LDT versioned key. See “Creating LDT Policies” on page 25.

   The following example shows an LDT production policy that transforms data encrypted by a non-versioned key (AES256) to data encrypted with a versioned key (LDT).



2. Make sure there is no application activity within the GuardPoint.

CAUTION

This step is critical. Do not skip it. Make sure there is no application activity within the GuardPoint.

3. Click **Unguard** to unguard the GuardPoint.
4. Guard the directory again using the new LDT policy. Use the steps in "Create an LDT GuardPoint for an Unguarded Directory" on page 39, but choose the policy that starts from the non-versioned/non-LDT key rather than a policy that starts from `clear_key`.

Rotating Encryption Keys (Rekey)

This section describes procedures related to encrypting with a new key, or rekeying. It covers the following topics:

Manual Key Rotation

If you need to rekey your GuardPoint prior to the expiration of the current key version, you can manually start a rekey process.

On Windows, you cannot rotate a key until all current data transformations that use the same key have completed, whether those transformations are an initial data transformation or a rekey. On Linux, you can rotate the key at any time. For details, see “Rotating Encryption Keys While a Rekey is in Progress (Relaunch)” on page 46.

1. To manually generate a new version of a key, open the Management Console and choose **Keys > Agent Keys > Keys.**
2. Click the name of the key you want to rotate.

   Note the hex value shown next to **Key Hash** before and after you rotate the key. The hash value represents the current key version associated with the key. After rotating the key, LDT generates a new version of the key, and the hash value changes.
3. Click **Rotate**.
4. After clicking Rotate, a Rotate Key pop-up displays. Select the *Rotation Reason* and click **Rotate**. This initiates the rekey process on all GuardPoints using the key.
5. Check the status of the GuardPoint in the Rekey Status column of the GuardPoint tab. For example, just after clicking Rotate, the status might be “Rekeying 4%”.

Once LDT transforms the data, you are finished.

Checking the Rekey Status

During a rekey, you can check the progress in the UI on the GuardPoint status page on the DSM.

GuardPoint status is not relayed to the DSM in real time. A delay of several minutes before the DSM displays events on the host is likely. When the number of GuardPoints on the managed host is high, for example 100+, the delay in relaying GuardPoint status is due to delays in scheduling and the execution of LDT operations on the managed host for GuardPoints.

When the host and DSM are configured for one-way communication, the delay is longer because LDT statistics are sent to the DSM once per hour, or each time a GuardPoint enable or disable status changes.

To check when the DSM last received a status update from the host, look at the Last Status Update time stamp in the GuardPoint status dialog. For information about how to display this dialog, see “Obtaining Statistics in the UI with GuardPoint Status” on page 83. For the most up-to-date statistics, inspect the host itself. To find out whether your host is configured for one-way communication, open the DSM Management Console, choose **Hosts** > **Hosts**, click the name of the host, and look for the FS Agent One Way Communication checkbox. To see the progress of a rekey and the estimated completion time:

1. Open the DSM Management Console.
2. Choose **Hosts** > **Hosts**.
3. Click the name of the host you want to check.
4. Click the **GuardPoint** tab.
5. The Rekey Status column shows the current status. The status is one of the following:

Rekey Status	Description
Error (Windows only)	Runtime error occurred. Check for LDT-ALERT messages. Typically occurs when Low disk space occurs or a volume got dismounted.

Rekey Status	Description
Exited (Windows only)	Exited an operation like renaming folder. LDT will be restarted in the next ~20 secs or as soon as the operation completes. On restart, it will go into scan phase if file/folder deleted/renamed.
Incomplete (Windows only)	LDT did not finish. The administrator must intervene to remove the obstacle so the rekey can finish. You can run the `voradmin ldt skip` command to obtain more information.
In-Progress	LDT is transforming each file one by one.
Not Started	LDT has not started. It will go into this state when a GuardPoint is added or a key is rotated or changed.
Paused	LDT is paused on the GuardPoint, because of the QoS schedule or a direct request from administrator.
Queued	When a key rotation is in progress and another rotation is initiated.
Rekeyed	GuardPoint has been transformed to the latest key version.
Rekeying	GuardPoint transformation is in process of transforming the data with the latest key.
Scanning	Linux: LDT started, it is scanning the GuardPoint and counting bytes, files to transform in the GP before starting transformation. Required for statistics and estimation. Windows: LDT is analyzing files in the GuardPoint before initial transformation, before a rekey, or following an interrupted rekey. The scan is required for statistics and estimation. LDT can scan multiple times on completing a traversal of the guarded directory. The scanning status does not include a percent complete estimate.
Starting	GuardPoint transformation is ready to begin, but has not yet begun.
Suspended	When LDT is suspended by user or due to QoS schedule.
Unknown	GuardPoint is not active on the host, so status could not be determined.

Obtaining Information About Keys Applied to Files

Key Report Option

In the following command, you can use the `report` option of the `voradmin` command to obtain information about all of the keys in use on the GuardPoint. The report lists all keys used in the GuardPoint. For each key, it gives the key name and key version number. It lists each unique key name and version combination only once, no matter how many files use the key.

The following example shows three keys used in the GuardPoint `/oxf-fs1/gp1`:

```
# voradmin ldt key report /oxf-fs1/gp1
```

System Response:

```
LDT_KEY1,1
LDT_KEY2,2
LDT_KEY3,5
```

For an overview of `voradmin ldt`, see "LDT Command-Line Administration: voradmin command" on page 75.

Key Map Option

In the following command, you can use the `map` option to obtain information about which files in a GuardPoint were transformed with a specific key.

- Replace `<key_name>` with a key name
- Replace `<guard path>` with the path of the GuardPoint where the key was used

 # voradmin ldt key [report|map] <*key_name*> <*guard path*>

Example:

 # voradmin ldt key map LDT_KEY2 /oxf-fs1/gp1

System Response:

```
/oxf-fs1/gp1/file12345.dat10
/oxf-fs1/gp1/file12345.dat10
/oxf-fs1/gp1/file12345.dat10
/oxf-fs1/gp1/file12345.dat10
/oxf-fs1/gp1/file12345.dat10
```

Keys without a version number are used by files in an exclusion key rule or files that have yet to undergo initial key rotation. Use `ldt key map` in conjunction with `ldt attr get` to determine if a file using a key without a version number is part of an exclusion key rule or awaiting initial key rotation.

Showing GuardPoints During Rekey (Linux)

Use the following command to display a list of known LDT metadata stores and any associated GuardPoints currently undergoing transformation.

 # voradmin ldt list all

System Response:

```
MDS_1:  type=file, nguards=0, name=/disk2/::vorm:mds::
        Guard Table: version 1 nentries 0
MDS_2:  type=file, nguards=0, name=/disk3/::vorm:mds::
        Guard Table: version 1 nentries 0
MDS_3:  type=file, nguards=0, name=/disk4/::vorm:mds::
        Guard Table: version 1 nentries 0
```

For an overview of `voradmin ldt`, see "LDT Command-Line Administration: voradmin command" on page 75.

Suspending and Resuming Rekey and/or Scan Phase

The QoS schedule specifies certain time windows when LDT operations must be stopped temporarily. However, you can also suspend LDT, or resume LDT at host level, at any time.

To suspend LDT through the Management Console UI:

1. Go to **Hosts** > **Hosts**.
2. Click the name of the host where you want to suspend or resume rekey.
3. Click the **GuardPoint** tab.
4. Click **Suspend/Resume Rekey** to suspend/resume LDT operations on the host.

Note: Disabling, or unguarding, an LDT GuardPoint where a rekey is in progress, automatically suspends the LDT rekey operation.

Note: You can also use the `voradmin suspend/resume` command in Windows Powershell to suspend and resume the rekey.

Note: On Windows, if a user uses the `voradmin ldt suspend` command to pause the rekey process, the LDT suspended state is not retained when the system reboots. If you want to retain the state after reboot, suspend the rekey from the DSM.

Note: Starting with VTE v.6.1.2 on Linux, you can pause LDT during the scan phase of Live Data Transformation. When paused during scan, LDT suspends operations that traverse through the namespace of GuardPoints in the scan phase of transformation. Suspending LDT during scan stops file lookup operations of LDT. This eliminates performance impact to I/O intensive production workloads, such as file serving type workloads, that access large number of files.

Automatic Suspend and Resume of LDT Operations Due to Insufficient Disk Space (Linux)

LDT requires adequate storage space headroom to perform LDT operations such as rekeying on GuardPoints. On Linux, if available storage space drops below the threshold required for LDT operations to continue, the LDT operations are automatically suspended. Once additional storage space is available in the file system, LDT operations automatically resume.

As available space approaches the threshold for automatic suspension, LDT sends an alert to the DSM to notify you that you should free up more storage space before LDT operations are suspended. The alert on the DSM is:

```
Low space on guard point [GuardPoint], increase free space or LDT will be suspended.
```

Behavior of Automatic Suspend and Resume of LDT Operations on ext4 File Systems

By default, ext4 file systems reserve a portion of the storage space for use only by privileged processes to prevent running out of storage space in file systems. In this situation, non-privileged processes are automatically blocked from writing to the file system until the free disk space level reaches the minimum threshold. As LDT operates in privileged mode, LDT operations continue without blockage even if the free disk space threshold is low. Because of this ext4 feature, LDT operations on ext4 file systems may not be suspended due to low available storage space, even when the `df` command reports storage is 100% allocated.

Rotating Encryption Keys While a Rekey is in Progress (Relaunch)

On Linux, if a key is rotated (either manually or automatically when a key version expires) while LDT is in progress on a GuardPoint, the key rotation is processed and queued, and the GuardPoint is marked for relaunch. Relaunch indicates the need to restart LDT after the current transformation completes. If the GuardPoint has been rekeyed and is flagged for relaunch, live data transformation launches as soon as the GuardPoint is enabled.

Files already undergoing transformation at the time of the new key rotation complete rekey to the key version already in progress. Files that start transformation after the new key rotation request are rekeyed to the new key version. Once the current LDT of the GuardPoint is complete, LDT automatically relaunches to transform any files that are not rekeyed to the latest key version.

For example:

```
# voradmin ldt attr get /oxf-fs1/gp1

LDT stats: version=2, rekey_status=rekeying,relaunch
        Number of times rekeyed:                1 time
        Rekey start time:                       2020/03/18 16:54:00
        Last rekey completion time:             2020/03/18 16:53:38
        Estimated rekey completion time:        0 days 0 hours 6 minutes
        Policy key version:                     344
        Data stats:
                total=9.8GB, rekeyed=0.0MB
                truncated=0.0MB, sparse=0.0MB
        File stats:
                total=1000, rekeyed=0, failed=0
                passed=0, skipped=0, created=0, removed=0, excluded=0
```

Note: Relaunch is supported on Linux only. It is not supported on Windows.

File System Operations

The following sections describe the file system operations that may require attention from the DSM administrator.

Renaming Files and Folders

On both Linux and Windows hosts, you can rename files during initial transformation or rekey operations.

On Linux hosts, however, you cannot rename folders (directories) in a GuardPoint during a rekey operation. Any attempt to rename a directory is rejected with the error EAGAIN (try again) while live data transformation is in progress including the period during which transformation is suspended.

On Windows hosts, you can rename folders during rekey. However, renaming a folder may trigger the rescan of the GuardPoint so the rekey process may reset or restart.

On Windows hosts for v6.1.0 and later, LDT behavior changes as follows:

- LDT stops if it is transforming the contents of a folder, and a user attempts to rename/move that folder.
- You can change the stopping behavior of LDT using the configuration parameter:
  - Using the Registry Editor, or the Windows command line, add a registry entry in: `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\Vmmgmt\ Parameters` for:

```
oxf_stop_on_rename
```

Registry Name	Values	Comments
`oxf_stop_on_rename`	0 - Disabled	LDT **does not** stop if there is contention.
	1 - Enabled	LDT **stops** if there is contention. Default: Enabled

- For target folder in rename operation, LDT is never stopped.

Deleting a File

When a file is removed before it is rekeyed, the file is not included in the total number of files transformed on the GuardPoint Status page. Any discrepancy between total number of files to transform and those transformed is due to the removal of files from GuardPoints during live data transformation.

The voradmin command provides more detailed file level statistics related to rekey operations on the host. You can run voradmin to get file level statistics:

```
# voradmin ldt stats <guard_path>
```

File Handling (Windows Only)

It is critical that you understand how the Live Data Transformation process handles read-only, binary (executable), NTFS encrypted and NTFS compressed files.

The LDT process is subjected to all of the File System policies and attributes set on the files. In some cases, this prevents LDT from encrypting a file. If users or applications are accessing files while LDT is in progress, LDT cannot change the attributes of the files and encrypt the file. It is critical that you understand how LDT handles various types of files:

- **NTFS Encryption and Compression**

  If NTFS encryption and compression is enabled on a file or folder, the LDT process cannot encrypt these files. To maintain the data coherency, LDT skips the encryption of the these files. These files display as “passthrough” files in the LDT statistics.
- **Read-Only Files**

  When LDT encounters read-only files, it rekeys the file by resetting the read-only attribute and then setting the attributes back again when the rekey completes. If a file is open, LDT skips this file.

  a. If the file is not opened, LDT changes the attributes of the file and stores the original attributes in the file metadata.

  b. LDT starts Rekey on this file.

  c. If a user requests to open a file for writing while rekey is in progress, access is denied. User can only open files for reading.

d. LDT restores the attributes once rekey is done.

- **Executable Files**

  If an executable is running, or files are exclusively locked by the application, the LDT process cannot encrypt those files as it is unable to acquire the required locks on the files. LDT skips these files and changes to the INCOMPLETE state.

Enabling GuardPoints in Read-Only mounted file systems (Linux)

Access to a GuardPoint enabled in a read-only mounted file system is restricted to read operations. You cannot modify data in such file systems, therefore, you cannot perform LDT operations on GuardPoints in read-only file systems. For this reason, LDT automatically suspends operations when GuardPoints are enabled on read-only file systems. LDT ignores all attempts to resume LDT operations until the underlying file system is remounted with read/write access.

WARNING

You must disable a GuardPoint before changing the read/write mount options of the underlying file system. After changing the mount options, you can re-enable the GuardPoint. LDT operations adapt to the read/write options of the underlying file system when you enable the GuardPoint. Changing the mount options while a GuardPoint is enabled is unsupported and may result in unexpected errors.

Copying Files Into a GuardPoint

If you copy a file into a GuardPoint without an Apply Key rule, make sure that the file was previously copied from the same GuardPoint, or a GuardPoint protected with the same policy/versioned key. A copy operation, without an Apply Key rule, is the same as a backup or restore of a file from/to an LDT protected GuardPoint.

VTE enforces key rules of an LDT policy while a GuardPoint is enabled. VTE cannot enforce the key rules while a GuardPoint is disabled. Modifying or adding data/files inside a disabled GuardPoint is not only unsupported, but it also results in unrecoverable data corruption.

WARNING

Do not add new files or modify existing files inside LDT protected GuardPoints while the GuardPoint is not enabled. This results in unrecoverable data corruption and/or files that cannot be accessed when the GuardPoint is enabled.

Behavior of Hard Links Inside and Outside of GuardPoints (Windows)

When using hard links on Windows, all the hard links to a file must be within the boundary of a GuardPoint and must use the same key. The following scenarios provide additional details:

- If hard links to the same file are inside a GuardPoint and outside a GuardPoint, the effect on the file depends on what process accesses which hard link first. If the hard link within the GuardPoint is opened first, the file is transformed. If the hard link outside the GuardPoint is opened first, the file won't be transformed.
- If hard links to the same file exist in different GuardPoints with different keys, the file will be corrupted.
- If hard links to the same file exist in the same GuardPoint but with different keys, such as if folder-based rules are used, there will be a conflict in the key.

Excluding Files or Directories from Rekey

You can exclude files or directories from the initial transformation and subsequent rekeys with exclusion key rules. For example, you can exclude a subset of non-secret files from a larger set of files that are encrypted with an LDT policy.

You set up exclusion key rules on a DSM. See the section, “Add Key Selection Rules” in the “Configuring Policies” chapter of the *DSM Administration Guide* (DSM 6.3.0 or higher).

Examples of Exclusion Key Rules

This section describes some examples of how to use an exclusion key rule.

Encrypt Files With Exclusion Property a Non-Versioned Key

The following exclusion key rule applies the non-versioned key, `Key_TextFiles`, to any new files that are created with a `*.txt` extension. Files with other extensions, such as `.doc` or `.zip`, that match other key rules in the same policy, may be encrypted with a different non-versioned key, a versioned key or `clear_key` (unencrypted).

Exclusion Key Rule: Resource set =*.txt, Key = Key_TextFiles

Note: Existing files with a `.txt` extension in the GuardPoint during initial data transformation are assumed to be encrypted with the same key specified key rule. Such files are not transformed during initial or subsequent rekey.

Exempt Excluded Files from Encryption (Set to clear_key)

The following exclusion key rule sets all files in the resource set `/oxf-fs1/gp1/Clear_Files_Folder` (Linux) or `\oxf-fs1\gp1\Clear_Files_Folder` (Windows) to `clear_key` (in other words, not encrypted). Files in other directories that match other key rules in the same policy may be encrypted. This could allow unrestricted access to the files in `/oxf-fs1/gp1/Clear_Files_Folder` (Linux) or `\oxf-fs1\gp1\Clear_Files_Folder` (Windows) while access may be restricted to files in parallel directories.

Linux Exclusion Key Rule: Resource set = `/oxf-fs1/gp1/Clear_Files_Folder`, Key = clear_key

Windows Exclusion Key Rule: Resource set = `\oxf-fs1\gp1\Clear_Files_Folder`, Key = clear_key

Requirements for Exclusion Key Rules

Keep in mind the following requirements when configuring exclusion key rules:

- Before adding an exclusion key rule to an existing policy, you must disable all GuardPoints protected with the policy. Log on to the DSM to disable the GuardPoint.
- You cannot choose a versioned key for the key in an exclusion key rule. Only non-versioned keys or `clear_key` (no encryption) are valid for exclusion key rules.
- Exclusion key rules can be specified in LDT policies created on DSM 6.3 or higher.
- Policies with exclusion key rules can be used on GuardPoints protected on hosts installed with VTE 6.2 or higher.
- Exclusion key rules should not be added to existing LDT policies used to protect GuardPoints on VTE 6.1.x or earlier hosts.

- All exclusion key rules must be above all LDT transformation key rules in the Key Selection Rules area in the policy.

Usage Notes and Limitations for Configuring Exclusion Key Rules

Keep in mind the information in the following sections when configuring an exclusion key rule.

Adding an Exclusion Key Rule to an Existing Policy with Versioned Keys (Linux)

When adding an exclusion key rule to an existing policy, the exclusion rule only applies to newly created files. Existing files that match the exclusion key rule remain encrypted with the same versioned key(s) specified in the non-exclusion key rule in the policy and will be rekeyed to the key in the exclusion key rule when the versioned key(s) rotates.

To force an existing file that matches an exclusion key rule to be transformed to the key in the exclusion key rule (non-versioned in this example), use one of the following methods:

- Rotate the versioned key specified in the policy to initiate rekey operations on the GuardPoint.
- Copy the existing file within the GuardPoint. The new file will be associated with the resource set in the exclusion key rule and will be encrypted with the non-versioned key. You can then delete the original file.

To perform a similar conversion on Windows, see “Changing a Folder or Files from Versioned to Non-Versioned Key (Windows)” on page 52.

Adding an Exclusion Key Rule That is Part of an Active GuardPoint (Linux)

To edit and/or add an exclusion key rule to an LDT policy, all GuardPoints using the policy must first be disabled before the new key rule can be added. See “Add Key Selection Rules” in the “Policies” chapter of the *DSM Administration Guide*.

Changing an Exclusion Key Rule That is Part of an Active GuardPoint (Windows)

Changes that you make to an exclusion key rule that is part of an existing policy in an active GuardPoint do not take effect until the GuardPoint that the exclusion key rule is part of is disabled and enabled again.

Conflicting Keys as the Result of Rename Operations

Do not attempt to move or rename a file encrypted with a versioned key to a name associated with an exclusion key rule with clear_key. If you attempt such a move or rename, the original file is unaffected but following error is output on Linux systems and a log entry is created on the DSM:

```
<command name>: setting attribute ‘user.::secfs:xattr:’ for ‘user.::secfs:xattr:’:
Invalid argument
<command name>: failed to close ‘<filename>’: Invalid argument
```

No error is displayed on Windows systems. The target moved or renamed file is corrupted and should be deleted. The target file is also flagged with the `xattr_error` flag on Linux and `Rekey Status Excluded` on Windows. This flag prevents subsequent read/write access to the file. You can check the LDT attributes for the presence of this flag. See “Determining if a File is Included in an Exclusion Key Rule (Linux)” on page 53 and “Determining if a File is Included in an Exclusion Key Rule (Windows)” on page 53.

Also, a log entry is sent to the DSM on Linux systems when this occurs. For example, if you moved the versioned file `/gp/foo.txt` into the GuardPoint `/gp/subdir/foo.txt` with an exclusion key rule that excludes matching files with the clear_key, the following log message would be created on the DSM:

```
[CGA] [WARN ] [29261] [CGS3268W] LDT-ALERT: encrypted data detected in filename
[foo.txt] inode [35720037] in guard point [/gp] under clear exclusion key rule
```

Overlapping Exclusion Key Rules

Multiple exclusion key rules in the same policy may overlap each other. For example on Linux, if the non-versioned key Key_A is associated with resource `/oxf-fs1/gp1/Folder_Enc_With_KEY_A` and the non-versioned key Key_B is associated with resource `*.txt`, placement of the file `/oxf-fs1/gp/Folder_Enc_With_KEY_A/foo.txt` overlaps the two key rules. In such a case, the first rule in the policy is enforced on `/oxf-fs1/gp/Folder_Enc_With_KEY_A/foo.txt` when the file is created and in subsequent file access.

On Windows, if the non-versioned key Key_A is associated with resource `c:\oxf-fs1\gp1\Folder_Enc_With_KEY_A` and the non-versioned key Key_B is associated with resource `*.txt`, then they would overlap on the file `c:\oxf-fs1\gp\Folder_Enc_With_KEY_A\foo.txt`. In such a case, the first rule in the policy is enforced on `c:\oxf-fs1\gp\Folder_Enc_With_KEY_A\foo.txt` when the file is created and in subsequent file access.

Caution About Applications That Create Temporary Files (Windows)

Some applications on Windows create a temporary file version of the original file when you open and modify a file. This behavior can affect how you implement exclusion key rules.

If you have an exclusion key rule that uses a file extension to exclude files that may be opened and modified by such an application, exclude the temporary file name extension also. If you don't exclude the temporary file, the temporary file may be encrypted by another policy that matches the temporary file extension. Then the original file, which is copied from the temporary file, will be unreadable. Also keep in mind that other applications that may create temporary files with the same file extension and consider what policies should affect those temporary files.

This situation can happen with Microsoft Office files such as the `.docx` files used by Microsoft Word. When you open and modify a `.docx` file, Word creates a `.tmp` file version of that file. So for Microsoft Word you should add `*.tmp` to the exclusion key rule resource set if you add `*.docx`.

Rename Operations Crossing Key Rules (Linux)

On Linux, when a rename operation crosses key rules, the rename operation copies the source file to a new file using the new name and removes the original file. If the source file is flagged for exclusion key rule property, the target new file is disassociated with the exclusion key rule if the new name no longer matches the resource set associated with exclusion key rule. For more information, see "About the Exclusion Attribute for Files Matching an Exclusion Key Rule" below.

Using Linked Files with Exclusion Key Rules (Linux)

On Linux, do not create multiple hard links to the same target file such that the pathname of each hard link is associated with a resource set of an exclusion key and the key rules have different keys. Accessing the file through the pathname of each hard link results in a different key to be applied to the file resulting in data corruption in the target file due to application of multiple keys to the same data.

If you must create hard links, be sure the pathnames of hard links and the target file are within the same resource set.

Changing a Folder or Files from Versioned to Non-Versioned Key (Windows)

Exclusion key rules provide a way to convert a subset of guarded files or the contents of a folder from being encrypted by a versioned key to being encrypted by a non-versioned key. After this conversion, the excluded files or directory contents will be encrypted at the last version of the versioned key but the encryption keys for those items will not rotate to a new version when the keys for other non-excluded files are rotated. In other words, the excluded files or folder contents will be encrypted by a non-versioned key.

Follow these steps to change selected files or folder contents to a non-versioned key:

1. In the DSM, clone the versioned key that is used in the policy that you plan to edit. Cloning a key creates a non-versioned copy of the existing version of the versioned key.
2. In the DSM, disable the GuardPoint. Disabling the GuardPoint is required before modifying a policy applied in that GuardPoint.
3. Configure one or more resource sets to match the files that you want to exclude. For example, the resource set `star-dot-text` could contain `*.txt` and the resource set `sales-folder` could contain `\sales\*`.
4. Add one or more exclusion key rules to convert matching files to a non-versioned key. In the following example, files matching `*.txt` and files in the `sales` folder (as defined in resource sets) will be converted from the current version of the versioned key `AES256_versioned` to `AES256_clone`, assuming `AES256_clone` is a clone of `AES256_versioned`.

Order	Resource	Current Key	Transformation Key	Exclusion Rule
1	star-dot-txt	AES256_clone	AES256_clone	Y
2	sales-folder	AES256_clone	AES256_clone	Y
3		clear_key	AES256_versioned	N

   Exclusion key rules must be ordered before other rules.
5. On the command line, run the following command to remove the LDT metadata from the files that you want to convert from versioned key to non-versioned key encryption:

   voradmin ldt attr delete <*path_to_files*>

   To recursively delete the LDT metadata from all files matching a pattern in all subfolders, use the following form, including `*` as a wildcard where needed:

   voradmin ldt attr delete <*path_to_files*> -filter <filename>.<extension>

Note: Be careful when deleting the LDT metadata from files. If you delete the metadata from a file that does not match an exclusion key rule policy, the file will be unreadable after the next rekey.

   Given the example exclusion key rule described in step 4, you would need to run this command on all files with the extension `.txt` (first example below) and on the files in `\sales` (second example below).

   voradmin ldt attr delete c:\gp1 -filter *.txt

   voradmin ldt attr delete c:\gp1\sales

   For more information about using `voradmin` on LDT metadata, run `voradmin ldt attr /?` on the command line.
6. In the DSM, re-enable the GuardPoint that includes the new exclusion key rule.
7. In the DSM, rotate the key for the policy that includes the new exclusion key rule.

Using the example exclusion key rule in step 4, after completing this procedure, files matching `*.txt` and files in the `sales` folder will have the exclusion attribute set and will be excluded from rekeying (see"About the Exclusion Attribute for Files Matching an Exclusion Key Rule" on page 53). Files not matching the exclusion key rule will be rekeyed to the next version of the `AES256_versioned` key.

To perform a similar conversion on Linux, see "Adding an Exclusion Key Rule to an Existing Policy with Versioned Keys (Linux)" on page 50.

About the Exclusion Attribute for Files Matching an Exclusion Key Rule

Files matching an exclusion key rule have the status `rekey_excluded` in the LDT attribute. For more information about LDT attributes, see "LDT Metadata in Extended Attributes" on page 59. To check for the exclusion attribute on a file, see "Determining if a File is Included in an Exclusion Key Rule (Linux)" or "Determining if a File is Included in an Exclusion Key Rule (Windows)" below.

The Exclusion Attribute is Persistent

Exclusion from rekey is a persistent property. A file excluded from rekey is not rekeyed regardless of changes that may seem to disassociate the file from the exclusion key rule. For example, if you rename a file to a new name within the same GuardPoint that no longer matches the exclusion key rule that the original name matched, the file with the new name retains the encryption type (non-versioned key or clear_key) of the exclusion key rule. To remove the exclusion attribute from a file you must copy the file rather than move or rename it (see "Removing the Exclusion Attribute From a File" below).

Determining if a File is Included in an Exclusion Key Rule (Linux)

Use the `voradmin` command to check whether a file is associated with an exclusion key rule. An excluded file will include the status `rekey_excluded` in the `voradmin` output.

- Open the command line interface on your system and type:

  `voradmin ldt attr get <path to file>`

  Example:

  `voradmin ldt attr get /oxf-fs1/gp1/foo.txt`

  System Response:

```
LDT attributes: rekeyed_size=0, rekey_status=rekey_excluded
```

Determining if a File is Included in an Exclusion Key Rule (Windows)

Use the `voradmin` command to check whether a file is associated with an exclusion key rule. An excluded file will include the attribute `Rekey Status Excluded` in the `voradmin` output.

- Open the command line interface on your system and type:

  `voradmin ldt attr get <path to file>`

  Example:

  `voradmin ldt attr get c:\gp1\foo.txt`

  System Response:

```
LDT attributes:
Rekey Status Excluded
Initial Rekeyed Size 9 Bytes
```

```
            Data Transformed 0 Bytes

            Key:
            Current Key Name/Version (Clear_Key)
            New Key Name/Version (Clear_Key)
```

Removing the Exclusion Attribute From a File

To disassociate a file from an exclusion key rule, you must copy the file to a new file not associated with the resource set of the same or another exclusion key rule as the source file. You can then remove the original file. The new file is created under the default key rule of whatever policy applies to the new file. Moving or renaming a file does not disassociate a file from an exclusion key rule (see "The Exclusion Attribute is Persistent" above).

For example, assume the following exclusion key rule where Key_TextFiles is a non-versioned key:

Exclusion Key Rule: Resource set =*.txt, Key = Key_TextFiles

If you copy the file `test1.txt` to `test1.foo`, `test1.foo` is created with whatever key is specified in the policy that matches the new file. The key for the new file could be a non-versioned key, versioned key, clear_key, or, no key at all if the new file is outside of a GuardPoint. The original file `test1.txt` remains unchanged and encrypted with the Key_TextFiles non-versioned key because the file remains in the exclusion key rule.

Rename and Restore Operations (Linux)

The effect of rename or backup/restore operations involving files associated or not associated with exclusion key rules is mixed and somewhat confusing. In some cases, the operations to restore or rename cause conflicts between the key associated with the source file and the key associated with the target file. For example, the result of a rename operation involving a source file not associated with an exclusion key rule and target file name associated with a resource set of an exclusion key rule is different from the result of the same operation when target file name is associated with the resource set of an exclusion key rule with clear_key.

Note: Be sure to review the effect of the operations below and avoid administrative operations that cross associations of files across multiple resource sets with conflicting key rules.

Backup/Restore

The table below illustrates the status and the key effect of restore operations involving mixed keys associated with source files from backup image and existing target files inside GuardPoint.

Key Applied to File in Backup	Key Applied to existing file in GuardPoint	Key Effect of Restore Operation on Existing file
Versioned	Versioned	Versioned
Versioned	Exclusion key	xattr_error (failed)
Versioned	Exclusion clear key	xattr_error (failed)
Exclusion key	Versioned	Exclusion key
Exclusion key	Exclusion key	Exclusion key
Exclusion key	Exclusion clear key	xattr_error (failed)
Exclusion clear key	Versioned	Exclusion clear key

Key Applied to File in Backup	Key Applied to existing file in GuardPoint	Key Effect of Restore Operation on Existing file
Exclusion clear key	Exclusion key	xattr_error (failed)
Exclusion clear key	Exclusion clear key	Exclusion clear key

The table below illustrates the status and the key effect of restore operations involving mixed keys associated with source files from backup image and new target files not present inside GuardPoint.

Key Applied to File in Backup	Key Applied to existing file in GuardPoint	Key Effect of Restore Operation on Existing file
Versioned	Versioned	Versioned
Versioned	Exclusion key	xattr_error (failed)
Versioned	Exclusion clear key	xattr_error (failed)
Exclusion key	Versioned	Exclusion key
Exclusion key	Exclusion key	Exclusion key
Exclusion key	Exclusion clear key	xattr_error (failed)
Exclusion clear key	Versioned	Exclusion clear key
Exclusion clear key	Exclusion key	xattr_error (failed)
Exclusion clear key	Exclusion clear key	Exclusion clear key

Rename Operation

The table below illustrates the status and the key effect of rename operations for different combinations of versioned, exclusion key, and exclusion clear key.

Key Applied to File in Backup	Key Applied to existing file in GuardPoint	Key Effect of Restore Operation on Existing file
Versioned	Versioned	Versioned
Versioned	Exclusion key	Exclusion key
Versioned	Exclusion clear key	xattr_error (failed)
Exclusion key	Versioned	Exclusion key
Exclusion key	Exclusion key	Exclusion key
Exclusion key	Exclusion clear key	Exclusion key
Exclusion clear key	Versioned	Exclusion clear key
Exclusion clear key	Exclusion key	Exclusion clear key
Exclusion clear key	Exclusion clear key	Exclusion clear key

Listing All Files Included in an Exclusion Key Rule (Linux)

Determining the files that match an exclusion key rule involves two steps. You list all the keys in a GuardPoint, choose the key that you're interested in, and then run a command to list the files that match that key. This process works for both standard key rules and exclusion key rules.

1. Decide on the GuardPoint that you want to check for excluded files. For example, for the GuardPoint `/oxf-fs1/gp1`, type the following on the command line:

   Example:

   ```
   voradmin ldt key report /oxf-fs1/gp1
   ```

   System Response:

   ```
   LDT_KEY1,1
   LDT_KEY2,2
   LDT_KEY3,5
   NON_VERSIONED_KEY
   ```

   The number after the comma is the key version number. See "Key Report Option" on page 43 for more information about the `voradmin` key report.

2. From the key report output, choose the key rule for which you want to list matching files.

   Example:

   ```
   # voradmin ldt key map NON_VERSIONED_KEY /oxf-fs1/gp1
   ```

   System Response:

   ```
   /oxf-fs1/gp1/file1.dat10
   /oxf-fs1/gp1/file2.dat10
   /oxf-fs1/gp1/file3.dat10
   /oxf-fs1/gp1/file4.dat10
   /oxf-fs1/gp1/file5.dat10
   ```

   See "Key Map Option" on page 44 for more information about the `voradmin` key map report.

Listing All Files Included in an Exclusion Key Rule (Windows)

Determining the files that match an exclusion key rule involves two steps. You list all the keys in a GuardPoint, choose the key that you're interested in, and then run a command to list the files that match that key. This process works for both standard key rules and exclusion key rules.

1. Decide on the GuardPoint that you want to check for excluded files. For example, for the GuardPoint `c:\gp1`, type the following on the command line:

   Example:

   ```
   voradmin ldt key report c:\gp1
   ```

   System Response:

   ```
   Keys used for GP, c:\gp1 :
   clear_key,0
   CS1-LDT-AES256,11
   ```

   The number after the comma is the key version number. The clear_key key is not versioned, so the version number is 0.

2. From the key report output, choose the key rule for which you want to list matching files.

   Example:

   ```
   voradmin ldt key map c:\gp1 clear_key,0
   ```

System Response:

```
Files rekey with key (clear_key,0)-
c:\gp1\file1.txt
c:\gp1\file2.txt
c:\gp1\file3.txt
c:\gp1\file4.txt
c:\gp1\file5.txt
```

Using LDT with SAP HANA Fibre Channel Systems (Linux Only)

You must add additional VTE commands to the HANA administrator entry.

- Using a text editor, edit the `/etc/sudoers` and add entries for `/usr/bin/voradmin` and `/usr/bin/vmsec`:

  Example:

  ```
  # hanadm ALL=NOPASSWD:
  ```

  System Response:

  ```
  /usr/bin/secfsd,/usr/bin/voradmin,/usr/bin/vmsec,/sbin/multipath,/sbin/
  multipathd,/etc/init.d/multipathd,/usr/bin/sg_persist,/bin/mount,/bin/
  umount,/bin/kill,/usr/bin/lsof,/sbin/vgchange,/sbin/vgscan
  ```

If you are using an ext3 file system, you must mount it with extended attributes.

- Using a text editor, edit the storage section of the `global.ini` file, type:

  ```
  partition_*_data__mountOptions = -o user_xattr

  partition_*_log__mountOptions = -o user_xattr
  ```

Note: See the ‘SAP HANA” chapter in the VTE Agent Installation and Configuration guide for more information.

If the policy specifies one key rule targeting a specific folder, other folders within GuardPoint outside of the specified folder are excluded from rekey.

An example is a policy with one key rule specifying a resource set that targets a folder within GuardPoint. Using /gp as a GuardPoint directory with three folders within /gp:

Chapter 5: LDT Administration

In addition to setting up data encryption, the LDT administrator must understand additional administration tasks.

This chapter contains the following sections:

LDT Metadata in Extended Attributes

An *extended attribute* is a name/value pair permanently associated with a file or directory stored in a file system. Vormetric Transparent Encryption (VTE) creates and maintains its own user extended attributes on LDT GuardPoint directories and files. The extended attributes are used to store metadata related to each file or directory that is protected using an LDT policy.

On Linux, LDT sets extended attributes on GuardPoint directories. The LDT attribute, of an LDT GuardPoint, stores the following metadata:

- Current key version
- Rekey status
- Rekey start and end times
- Estimated completion time
- Total amount of data transformed
- Total number of files transformed
- Current key signature and applied key signature

On both Linux and Windows, LDT sets extended attributes on files. The LDT attribute, of a file, stores the following metadata:

- Name of the current key
- Name of the versioned key

  Also referred to in the Management Console UI and product documentation as the new key or the transformation key
- Version number of the versioned key
- LDT rekey status of the file

In most cases, the current and new key names are the same. The exception is during initial transformation from a legacy policy to an LDT policy, when the file has been encrypted with the current key and is being transformed to the current version of the transformation key.

Note: Before you set up a GuardPoint for LDT, ensure that there is sufficient disk space available in your file system for LDT metadata. The amount of disk space you need depends on the number of files in your GuardPoint. For more information about the disk space requirement, see "Planning for LDT Attribute Storage" on page 62.

The state of a file changes during LDT operations. The extended attributes are continually updated to reflect the current file status, which falls into one of the following categories:

- Rekeyed to the current version of the key.
- Rekeyed to the previous version of the key, or the initial key state (before the first LDT rekey has been performed).
- Partially rekeyed, where some regions of the file are rekeyed to the new key version and other regions are still keyed to the previous key version or the initial key.

Listing Extended Attributes

You can list extended attributes by using native operating system commands, or system calls. As part of GuardPoint administration, VTE can modify or delete extended attributes.

In Linux, LDT attributes are set on GuardPoint directories and regular files in GuardPoint directories protected with LDT policies. The VTE extended attribute name is `::secfs:xattr:`.

The following example uses the native Linux operating system command `attr` to display the LDT attribute for the GuardPoint `/oxf-fs1/gp1` and the file `/oxf-fs1/gp1/File_1.txt`.

Example 1:

```
# attr -l /oxf-fs1/gp1/File_1.txt
```

System Response:

```
Attribute "::secfs:xattr:" has a 1044 byte value for /oxf-fs1/gp1/
File_1.txt
Attribute "selinux" has a 37 byte value for /oxf-fs1/gp1/File_1.txt
```

Example 2:

```
# attr -l /oxf-fs1/gp1
```

System Response:

```
Attribute "::secfs:xattr:" has a 1044 byte value for /oxf-fs1/gp1
Attribute "selinux" has a 37 byte value for /oxf-fs1/gp1
```

Example 3:

In the following example, the file `/oxf-fs1/gp1/File_1.txt` has the same name for current and new keys at the same key version. In the following example, if the versioned key LDT_KEY is at version 755, the file is rekeyed to the latest key version under the LDT policy.

```
# voradmin ldt attr get /oxf-fs1/gp1/File_1.txt
```

System Response:

```
LDT attributes: rekeyed_size=4096, rekey_status=none
 Key:    name=LDT_KEY, version=755
```

Example 4: (Linux)

The following is example of an LDT attribute on a GuardPoint directory on Linux

```
# voradmin ldt attr get /oxf-fs1/gp1
```

```
LDT stats: version=1, rekey_status=rekeying
   Number of times rekeyed:          3 times
   Rekey start time:                 2018/08/04 16:24:45
   Last rekey completion time:       2018/07/04 16:24:04
   Estimated rekey completion time:  N/A
   Policy key version:               2043
   Data stats:
      total=3.3GB, rekeyed=1.5GB, truncated=0.0MB
   File stats:
      total=4307, rekeyed=1181,
      passed=2, skipped=0, created=0, removed=0
```

Example 5: (Windows)

The attribute for the GuardPoint `c:\GP 1` contains the status (rekeyed) and statistics specific to the GuardPoint and LDT. Following is sample output of voradmin command on Windows for statistics on a file:

```
C:\Users\Administrator> voradmin ldt attr get c:\GP\Test.txt
```

System Response:

```
LDT attributes:
    Rekey Status           Rekeyed
    Initial Rekeyed Size   10 Bytes
Key:
    Key Name/Version       (LDT_KEY, 28)
```

The attribute for GuardPoint C:\GP contains the status (rekeyed) and statistics specific to the GuardPoint and LDT:

```
C:\> voradmin ldt attr get c:\gp\
```

System Response:

```
Live Data Transformation Stats
------------------------------
    Rekey Status                                LDT_ST_REKEYED
    Last rekey completion time        10/2/2017 4:26:50
    Rekey Start time                            10/2/2017 4:26:17
    Estimated rekey completion time   000:00:00

File Stats:
    Total      444
    Rekeyed    444
    Skipped    0
    Errored    0
    Passed     0
    Removed    0

Data Stats:
    Total      11 GB (12649143417 Bytes)
```

```
                Rekeyed    11 GB (12649143417 Bytes)
                Truncated  0 Bytes
```

MDS File (Linux)

In addition to LDT attributes, the LDT process on Linux requires persistent storage for additional metadata related to encrypting, or rekeying, files in GuardPoints. LDT allocates the storage space as soon as the live data transformation process starts on a GuardPoint. It maintains this storage space during the entire transformation process, until the GuardPoint is completely transformed.

Storage for this metadata is allocated and managed in a special file, called the MDS (metadata store) file. The MDS file resides inside a GuardPoint directory so each GuardPoint has its own MDS file.

The MDS file is a VTE protected file with the name `::vorm:mds::`. For example:

```
# ls -l /oxf-fs1/gp1/::vorm:mds::
```

System Response:

```
 -rwxr-xr-x. 1 root  root 31754474496 Dec  8 09:09 /oxf-fs1/::vorm:mds::
```

```
# du -B 1024 /oxf-fs1/gp1/::vorm:mds::
```

System Response:

```
 25056   /oxf-fs1/::vorm:mds::
```

As shown above, the MDS file is sparse. In the example, the file size is approximately 30GB, however the file is allocated with approximately 25MB of disk storage. LDT automatically creates the MDS file the first time the LDT process starts on any GuardPoint in the file system namespace. It populates the MDS file with all of the metadata for the GuardPoint at the beginning of the LDT process. Disk space allocated to the MDS file is freed and the MDS file in the GuardPoint directory is removed when the LDT process completes on the GuardPoint.

WARNING

The MDS file is protected. You cannot remove it unless the administrator runs the voradmin command to manually remove the MDS file once it is no longer needed. See “Deleting LDT Metadata (Linux)” on page 81 for more information.

LDT automatically allocates and deallocates disk space for the MDS file as part of the LDT process. Deallocation of disk space for a GuardPoint does not change the MDS file size, although it frees the disk blocks. MDS files are sparse and very large in size. With no GuardPoint undergoing LDT, the minimum allocation of an MDS file is approximately 25MB.

Planning for LDT Attribute Storage

Before a GuardPoint is enabled for Live Data Transformation, make sure that there is sufficient free disk space in the file system to which the GuardPoint belongs. Free space is required for LDT attributes (VTE extended attributes) and (in Linux) metadata in the MDS file. LDT attributes are created during initial encryption and are never freed until the GuardPoint is permanently unguarded and removed from the protection of an LDT policy. In contrast, disk space for metadata in the MDS file is temporary, kept only during the live transformation process.

When planning how much free disk space to reserve for LDT on a GuardPoint, consider the following:

- Number of files in the GuardPoint

- (Linux) Average length of absolute pathnames of files in the GuardPoint

The LDT process pre-allocates disk space for the Linux MDS file based on a minimum of 200K files with an average pathname of 1024 bytes per GuardPoint. The minimum space amounts to 325MB of disk space for the MDS file for each GuardPoint, even if file count is very low. (In Windows, LDT reserves the space when the file is rekeyed.)

Using voradmin To Estimate Disk Space Required for LDT (Linux)

In Linux, you can use the `voradmin ldt space` command to estimate the amount of disk space required for LDT attributes and the MDS file. The result is rounded to the nearest MB. The syntax of the command is:

```
# voradmin ldt space <guard path>
```

The following example shows how to estimate the required disk space to perform Live Data Transformation on 1501 files in the GuardPoint `/oxf-fs1/gp1`. Estimate the disk space before protecting `/oxf-fs1/gp1` using an LDT policy:

Example:

```
# voradmin ldt space /oxf-fs1/gp1
```

System Response:

```
/oxf-fs1/gp1: found 1501 files without LDT extended attributes
LDT disk space requirements: total 169MB (LDT attributes=6MB, MDS=163MB)
```

The voradmin command reports that 1501 files in `/oxf-fs1/gp1` without LDT attributes. These files are new to LDT. 6MB of space is required for the LDT extended attributes and 163MB for metadata in the MDS file.

The following example shows the output of the same command after encryption completes. This estimates the additional disk space needed for the next LDT rekey operation:

Example:

```
# voradmin ldt space /oxf-fs1/gp1
```

System Response:

```
/oxf-fs1/gp1: found 0 files without LDT extended attributes
LDT disk space requirements: total 163MB (LDT attributes=0MB, MDS=163MB)
```

The voradmin command still reports on the same 1501 files, but because encryption has occurred using LDT, the files all have their LDT attributes. No additional space is required for LDT extended attributes on the next run. However, since the MDS file is transient and deletes after it finishes encrypting the GuardPoint, it requires additional space for the next key rotation.

Note: Windows estimates space using the following equation:
Permanent Space = Number of Files * 4K
Temporary Space = 128K * (Number of CPU * 2)

Displaying Metadata

Use the following command to see LDT file attributes or GuardPoint attributes:

```
# voradmin ldt attr get [<file name path> | <guard path>]
```

For an overview of `voradmin ldt`, see “LDT Command-Line Administration: voradmin command” on page 75.

Verifying Metadata (Windows only)

Use the following command to verify if the metadata is corrupt:

```
# voradmin ldt attr verify [<file name path> | <guard path>]
```

For an overview of `voradmin ldt`, see “LDT Command-Line Administration: voradmin command” on page 75.

DFSR and Replication (Windows)

The Distributed File System Replication (DFSR) service is a multi-master replication engine used to maintain synchronized folders on multiple servers. Replicating data to multiple servers increases data availability and provides users in remote sites with fast, reliable access to files.

If you are creating GuardPoints in a DFSR environment, you must first add the DFSR private folder guard path to the exclusion list in the Windows Registry. LDT should not attempt data transformation on this read-only directory.

1. Using the Registry Editor, or the Windows command line, add a registry entry in: `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\Vmmgmt\Parameters` named "LDTExclusionGPList" of type REG_MULTI_SZ.
2. Add the `DfsrPrivate` directory path to the `LDTExclusionGPList`.

   For example, if the DFSR private directory path is `D:\data\DfsrPrivate`, add this string in LDTExclusionGPList.
3. Reboot the system to make the change take effect.

Note: LDT does not perform a rekey on the DFSR private directory. Its rekey status is always ”N/A”.

If an application, or user, is performing a rename or a delete folder operation inside a GuardPoint with an LDT policy, this may restart the rekey process. Files which are already rekeyed will not be rekeyed again.

- If a rename or a delete operation is already performed on rekeyed files/folder, then the rekey process does not restart.
- If a rename or delete operation is performed in a folder where a rekey is in progress, LDT needs to stop the rekey process and restart the rekey again.

Multi-Node Configuration and Operation

This section describes what to expect at runtime when LDT operates in a multi-node cluster environment.

LDT Behavior on Failover

In a multi-node cluster, LDT runs in active-passive mode. A GuardPoint is only enabled on the active node, and LDT only runs on the active node. On the DSM Management Console, you can only view statistics for the active host.

If the active host fails over, and the standby host takes control, LDT also fails over to the second host. On the DSM Management Console, LDT statistics now appear on the other host.

Veritas Cluster File System

In an HA environment with Veritas Cluster Server, manual failover of service to a stand-by host may require manual intervention. If any GuardPoint is in the scan phase at the time of the manual failover, the failover process fails with the error "UNABLE to OFFLINE". The reason for the failure may be the scan state of a GuardPoint that cannot be suspended. LDT suspends LDT operations during the rekey state of transformation. Because LDT scan operations cannot be suspended, the Veritas Cluster Server fails to suspend LDT on GuardPoints in the scan phase of LDT operation. This state of LDT operation prevents Veritas Cluster Server to manually offline the service for fail-over.

Backing Up and Restoring LDT GuardPoints

This section describes procedures and considerations related to backing up and restoring data in LDT GuardPoints.

In addition to the files in a GuardPoint, LDT stores metadata in extended file attributes. These LDT attributes contain information that is required for decrypting the files and for the proper operation of LDT. Therefore, it is critical that your backup application also backs up the extended attributes of the files along with file data.

Clear Text Backup and Restore

A policy that applies a Security Rule with the Apply Key effect on backup/restore operations does not require any special rules for data access by backup applications. Under such a policy, backup applications always read clear data and store clear data in backup media. The backup application is not required to back up the LDT extended attributes, and a QoS schedule is not required to suspend LDT during backups.

Encrypted Backup and Restore

When a policy does not enforce a Security Rule with the Apply Key effect on backup/restore operations, the policy does not decrypt data on IO operations from that backup application. Under such a policy, the backup application stores encrypted data and the LDT extended attributes of the file on the backup media.

In Linux, LDT operations must be suspended during backup. Suspending LDT completes the ongoing rekey operations on regions of files before starting the backup. During live transformation, LDT first preserves those regions of a file to be rekeyed in the MDS file. Then it updates some of the metadata in order to update the status of the data preserved in the MDS file in preparation for the rekey. Then it starts rekeying and updating those regions in the underlying file.

Suspending LDT waits for ongoing rekey operations to complete, and saves the metadata in the LDT extended attribute section of the MDS file. Suspending LDT ensures that the rekey status stored in the LDT extended attribute accurately reflects the rekey status of the data in the entire file during backup.

Note: This requirement does not apply to LDT for Windows.

The following table summarizes the state of the data in files in backup media:

Table 5-1: State of data in files in backup media

Live Data Transformation State of File	Security Policy	Backup Metadata or Alternate Data Streams Along with File data	Effect
Not rekeyed	Permit	Yes	Data in backup may be in clear format or encrypted with older key version.
Rekey in progress	Permit	Yes	File in backup storage is partially rekeyed. Some parts are in clear format or encrypted with older key version, and other parts are encrypted with current key version.
Rekey complete	Permit	Yes	File in backup storage is in the encrypted format with new key version.

LDT policy for encrypted backup and restore

Suppose you have an LDT policy allowing a backup user, or a backup process, to perform an encrypted backup. The backup user, or the backup process, reads encrypted data from files and stores the same encrypted data in backup media.

For example, suppose you wanted to back up GuardPoint `/oxf-fs1/gp1` protected by **My_LDT_Policy_1**. The key version before the key rotation is 8. The following steps occur:

1. Key version 8 expires and is rotated to version 9.
2. Rotating the encryption key triggers LDT to start running on the data under the GuardPoint.
3. The QoS schedule suspends LDT because a backup process is running.
4. The backup process begins, runs and later ends.
5. The QoS schedule resumes LDT.
6. LDT on the GuardPoint completes.

Note: Although the backup/restore scenarios and examples described in the rest of this section are specific to LDT on Linux platform, the concepts also apply to LDT on the Windows platform.

Consider the state of the three files in the GuardPoint during the LDT process, right after LDT is suspended in preparation for the backup. You can obtain the state of each file through the `voradmin ldt attr get` command, which examines the LDT state of each file captured in the LDT extended attributes of those files. (For an overview of `voradmin ldt`, see "LDT Command-Line Administration: voradmin command" on page 75).

- *File_1.dat* is rekeyed/encrypted to completion. The applied and new key versions are at version 9 of the key.

```
# voradmin ldt attr get /oxf-fs1/gp1/data_files /file_1.dat
 LDT attributes: rekeyed_size=1440743424, rekey_status=none
     Key:    name=LDT_AES256_KEY, version=9
```

- *File_2.dat* is partially rekeyed/encrypted. The applied key version is at version 8 and the transformation key is at version 9.

```
# voradmin ldt attr get /oxf-fs1/gp1/data_files/file_2.dat
 LDT attributes: rekeyed_size=1440743424, rekey_status=rekeying,state_saved
     Current Key:    name=LDT_AES256_KEY, version=8
     New Key:        name=LDT_AES256_KEY, version=9
```

- *File_3.dat* has not been rekeyed/encrypted. The applied key and transformation key are both at version 8, which is the version before the current key rotation.

```
# voradmin ldt attr get /oxf-fs1/gp1/data_files/file_3.dat
 LDT attributes: rekeyed_size=1440743424, rekey_status=none
     Key:    name=LDT_AES256_KEY, version=8
```

While LDT is suspended on the GuardPoint, the backup process starts and archives these three files, including extended attributes, in the backup media.

```
# my_backup --preserve=xattr /oxf-fs1/gp1/data_files/file_1.dat \
        /backup-media/oxf-fs1/gp1/data_files/file_1.dat
# my_backup --preserve=xattr /oxf-fs1/gp1/data_files/file_2.dat \
                /backup-media/oxf-fs1/gp1/data_files/file_2.dat
# my_backup --preserve=xattr /oxf-fs1/gp1/data_files/file_3.dat \
                /backup-media/oxf-fs1/gp1/data_files/file_3.dat
```

Note: On Linux platforms, you must perform backups on a GuardPoint during periods when LDT is not actively transforming data in the GuardPoint. You must schedule backups in conjunction with LDT schedules or during periods when LDT is paused. Make sure to coordinate with the backup administrator and schedule accordingly.

Note: If the preferred method of backup is to create device level snapshots of the file system on the managed host or in the storage array subsystem, you must ensure that the schedule for creating snapshots pauses LDT before a snapshot is created.

Backup/Restore of Metadata Store File (MDS) in GuardPoints Undergoing Rekey

Backing up an entire GuardPoint using commands such as, `cp`, `tar`, or `rsync`, or even commercial backup products that sweep the file system namespace for backing up files is supported. This method of backup performed while live transformation is in progress over GuardPoint poses some challenges when files in GuardPoint are backed up in encrypted format. The challenge with this method of backup is existence of the LDT Metadata Store file (MDS) in GuardPoint during backup or restore process and strict protection enforced on MDS files preventing deletion or modification to MDS files. Second, your backup utility must backup and restore extended attributes of files, and because LDT extended attributes are also protected similar to MDS files, your backup must be enabled to restore extended attribute of files. To overcome both restrictions, your policy on GuardPoint must include a security rule without Apply Key Effect to enable your backup utility to replace MDS file and LDT extended attribute of files as part of backup/restore operations. Additionally, your backup utility must be signed with a signature set from the DSM for added security, and be executed with the option to preserve extended attribute options available with the standard Linux utilities.

The following table lists the supported backup utilities, required options to preserve extended attributes, and supported versions of the utilities.

Command	Required options	Supported version
`cp`	`--preserve=all`	OS default
`tar`	`--xattrs`	OS default
`rsync`	`-vapIXWP --inplace`	OS default
netbackup	overwrite existing files	v7.6.1 and v7.7.3

Starting with v6.1.3 of the VTE agent, you can also backup and restore GuardPoints data including MDS file, if the requirements listed above on the backup utility are satisfied. GuardPoints associated with an MDS file located outside of GuardPoint namespace cannot be backed up or restore using this method of backup.

Note: You must suspend LDT on GuardPoint before performing backup or restore.

You can check your GuardPoint to determine if you can use this method of backup. If the directory of your GuardPoint is a mount point, MDS files reside inside your GuardPoint and this method of backup can be used for backing up or restoring your GuardPoint. However, GuardPoint directories below file system mount points must be checked and verified to use this method of backup. To verify, you can run the `voradmin` command to determine the MDS file associated your GuardPoint. As an example:

```
# voradmin ldt list all
```

System Response:

```
MDS_1:  type=file, nguards=1, name=/oxf-fs1/gp1/::vorm:mds::

Guard Table: version 1 nentries 1
Guard 0: type=GP, state=REKEYING SUSPENDED (vadm), flags=GP LOCKED,
gp=/oxf-fs1/gp1
File List: count 4308
```

The report on `/oxf-fs1/gp1` GuardPoint indicates association of the MDS file `/oxf-fs1/gp1/::vorm:mds::` with the GuardPoint. As the MDS file resides inside the GuardPoint directory, you can use this method of backup to backup and restore this GuardPoint.

Before restoring the GuardPoint, you must suspend LDT operations on the GuardPoint if live transformation is in progress. Failure to do so will fail to restore the MDS file in the backup image. Failure to restore MDS file affects partially rekeyed files restored to the GuardPoint. In such a situation the restored data is invalid, and you must remove all the files restored to the GuardPoint and repeat the restore operation after suspending LDT.

Restoring a GuardPoint from a backup

To properly restore GuardPoint along with the MDS file, the following steps must be done in order:

1. Verify if live transformation is in progress on the GuardPoint and suspend the LDT operations. (Skip this step if no live transformation is occurring.)
2. With the same tool that was used for backup, restore to the GuardPoint.

3. Once restore is complete, GuardPoint needs to be disabled and enabled again. Run the `voradmin` command as shown below to determine how the GuardPoint was suspended at the time of backup. For manual GuardPoint, run the `secfsd` command as shown below. For auto-guards, disable and enable GuardPoints on DSM.

```
# secfsd -unguard <guardpoint>

# secfsd -guard <guardpoint>
```

4. Since GuardPoint was in suspended state during the time of backup, it will be restored in suspended state. You must resume LDT to complete live data transformation on the restored GuardPoint. You must resume LDT on the DSM if LDT was suspended on the DSM at the time of backup, otherwise you will resume LDT using `voradmin` command. Run the `voradmin` command below to determine if LDT must be resumed on DSM or through `voradmin`. If the tag string next to SUSPENDED state of GuardPoint is `vadm`, run `voradmin` command to resume LDT, otherwise the tag value is `qos` and LDT must be resumed on DSM. In the example below, GuardPoint `/oxf-fs1/gp1` was suspended using `voradmin`, and it must be resumed after restore using `voradmin` command.

```
# voradmin ldt list all

MDS_1:  type=file, nguards=1, name=/oxf-fs1/gp1/::vorm:mds::
        Guard Table: version 1 nentries 1
       Guard 0: type=GP, state=REKEYING SUSPENDED (vadm), flags=GP LOCKED,
gp=/oxf-fs1/gp1
                File List: count 4308

# voradmin ldt resume /oxf-fs1/gp1
```

5. Wait for LDT completion on GuardPoint.
6. We strongly recommend to disable and enable GuardPoint once more

The restore is now complete and files in your GuardPoint can be accessed.

The VTE agent sends the following alert message to DSM if the restore operation is rejected:

```
LDT-ALERT: Restore of LDT protected file <GuardPoint> not allowed by policy
```

Potential Warnings during restore operation

When using `cp` for backup/restore, the `cp` command may report a failed attempt to preserve permissions on the Metadata Store File (MDS) during a restore. If you encounter the below message, continue to proceed with the restore steps since this will not affect the MDS file or dataset that is being restored.

```
cp: preserving permissions for ‘/oxf-fs1/gp1/::vorm:mds::’: Permission denied
```

When using `rsync` for backup/restore, the `rsync` command may report a failed attempt to set extended attribute when restoring the MDS file on system with selinux configured in enforced mode. If user encounters the below message, continue to proceed with the restore steps since this will not affect the MDS file or dataset that is being restored.

```
rsync: copy_xattrs: lsetxattr(""/oxf-fs1/gp1/::vorm:mds::"","security.selinux")
failed: Permission denied (13)
```

```
rsync: rsync_xal_set: lsetxattr(""/oxf-fs1/gp1/::vorm:mds::"","security.selinux")
failed: Permission denied (13)
```

Restore an encrypted backup

This section illustrates restoration of the three files from the backup media to the same GuardPoint, */oxf-fs1/gp1*. The files are restored to a different directory under the GuardPoint.

Restore a file fully rekeyed to the latest key version

Recall that *file_1.dat* was archived in the backup media when it was fully rekeyed to version 9 of the key. As the current version of the key is also 9, *file_1.dat* is restored from backup without any changes. After restoring the file, the state of the restored file and its applied and current key versions remain unchanged, as compared to the original file that was backed up.

```
# my_backup --preserve=xattr /backup-media/oxf-fs1/gp1/data_files \ /file_1.dat /
oxf-fs1/gp1/restored_files/file_1.dat

# voradmin ldt attr get /oxf-fs1/gp1/restored_files/file_1.dat
```

System Response:

```
   LDT attributes: rekeyed_size=1440743424, rekey_status=none
      Key:    name=LDT_AES256_KEY, version=9
```

```
# voradmin ldt attr get /oxf-fs1/gp1/data_files/file_1.dat
```

System Response:

```
   LDT attributes: rekeyed_size=1440743424, rekey_status=none
      Key:    name=LDT_AES256_KEY, version=9
```

Restore a partially rekeyed/encrypted file

Recall that *file_2.dat*, archived in the backup media, was partially rekeyed between versions 8 and 9. As the current version of the key was 9 at the time of backup, *file_2.dat* is restored to the GuardPoint with the same version of the key from the time of backup. The file is flagged for *lazy rekey*, meaning that a background rekey operation is scheduled to transform the file to the latest key version the next time an application tries to access the file.

At the completion of restoration, the file is fully transformed to the key version (v9). The key version is also the latest one in the policy. Although the file is flagged for lazy rekey (LAZY_RK), the file does not need to be transformed to the latest key version because it's already there. Had the file been partially rekeyed from version 7 to version 8 of the key at the time of backup, the restored file would have completed rekeying to version 8 at the end of the restoration. Therefore, the LAZY_RK flag would initiate a background transformation to update the key version to the latest key version when the file is accessed.

If this file is not accessed by any application, the file remains unchanged in the GuardPoint. It is not transformed to the latest key version. To trigger a rekey, either re-push the LDT policy from the DSM, or access the file with an application, such as a file explorer.

```
# my_backup --preserve=xattr /backup-media/oxf-fs1/gp1/data_files/\ file_1.dat /
oxf-fs1/gp1/restored_files/file_2.dat

# voradmin ldt attr get /oxf-fs1/gp1/restored_files/file_2.dat
```

System Response:

```
 LDT attributes: rekeyed_size=1440743424, rekey_status=lazy_rekey
      Key:    name=LDT_AES256_KEY, version=9
```

```
# voradmin ldt attr get /oxf-fs1/gp1/restored_files/file_2.dat
```

System Response:

```
  LDT attributes: rekeyed_size=1440743424, rekey_status=none
      Key:    name=LDT_AES256_KEY, version=9
```

Restore a file not rekeyed/encrypted with an older key version

Recall *file_3.dat* was archived in the backup media when it was keyed to version 8 of the key, one version below the latest version at the time of backup. At completion of the restoration, *file_3.dat* is restored from backup to the same version, version 8, that it was keyed to when it was backed up.

However, the file is flagged for lazy rekey. After restoring the file, it is keyed to version 8 and flagged for lazy rekey (LAZY_RK). The file is rekeyed to the latest key version, version 9, as soon as an application accesses the file. If this file is not accessed by any application, the file remains unchanged in the GuardPoint.

```
# my_backup --preserve=xattr /backup-media/oxf-fs1/gp1/data_files \ /file_3.dat /
oxf-fs1/gp1/restored_files/file_3.dat
```

```
# voradmin ldt attr get /oxf-fs1/gp1/restored_files/file_3.dat
```

System Response:

```
      LDT attributes: rekeyed_size=1440743424, rekey_status=lazy_rekey
      Key:    name=LDT_AES256_KEY, version=8
```

```
# sum /oxf-fs1/gp1/restored_files/file_3.dat
```

System Response:

```
      39994 1406976
```

```
# voradmin ldt attr get /oxf-fs1/gp1/restored_files/file_3.dat
```

System Response:

```
      LDT attributes: rekeyed_size=1440743424, rekey_status=none
      Key:    name=LDT_AES256_KEY, version=9
```

Restoring Non-LDT Backup Data to an LDT GuardPoint

This section describes how to restore data encrypted with a non-versioned key to an LDT GuardPoint.

If the backup was performed with the Apply Key effect, the backup files are in clear text. Simply restore the clear text files to the LDT GuardPoint with the Apply Key effect. All files will be encrypted with the versioned key.

If the backup of the non-LDT GuardPoint was performed without the Apply Key effect, the backup is encrypted, and you must do the following:

Note: The following example is for a manual guarding. The steps may differ slighly if your GuardPoint is configured for auto guard.

1. Create a temporary directory for restoring the files, type:

```
# mkdir -p /oxf-fs1/tmp_restore
```

2. Restore the encrypted backup files into the temporary directory, type:

```
# cp -pr /backup-media/oxf-fs1/gp1/data_files/* /oxf-fs1/tmp_restore
```

3. Create a Standard Policy with the Apply Key effect for all operations, using the same key as the policy applied on the GuardPoint at the time of backup.
4. Create and enable a new GuardPoint for the temporary directory using the Standard Policy just created.

```
# secfsd -guard /oxf-fs1/tmp_restore
```

5. Ensure that the temporary GuardPoint and LDT GuardPoint are both enabled.

```
# secfsd -status guard
```

System Response:

```
GuardPoint           Policy      Type       ConfigState  Status       Reason
----------           ------      ----       -----------  ------       ------
/oxf-fs1/gp1          LDT_AES256  manual      guarded      guarded      N/A
/oxf-fs1/tmp_restore  AES256      manual      guarded      guarded      N/A
```

6. Move the restored files from the temporary folder to the GuardPoint enabled with the LDT policy. The VTE agent encrypts the files in the LDT GuardPoint using the current key version in effect for the LDT policy.

```
# mv /oxf-fs1/tmp_restore/* /oxf-fs1/gp1
```

7. Disable the temporary GuardPoint and remove the temporary restore directory.

```
# secfsd -unguard /oxf-fs1/tmp_restore

# rm -fr /oxf-fs1/tmp_restore
```

8. Delete the temporary GuardPoint on the DSM.

Using fsfreeze (Linux only)

If you use the `fsfreeze` command to quiesce access to the file system before creating a snapshot, refer to the fsfreeze section in VTE Admin Guide in the Linux Utilities chapter on how to run the fsfreeze command to quiesce access to both the file system and the GuardPoint(s).

WARNING

Do not use LDT schedules and do not pause LDT to align backup schedules with LDT. Instead, use the `fsfreeze` command.

- Running `fsfreeze -f` on the GuardPoint directory pauses LDT operations in-progress and freezes access to both the GuardPoint and the underlying file system.
- Running `freeze -u` reverses freeze -f, allowing access to underlying file system and resuming LDT operations.

LDT Backups Using a File System or Storage-Level Snapshot Tool

You can make a file system snapshot using a Logical Volume Manager service or mirroring/splitting storage level LUNs of a file system inside the storage subsystem. LDT does not have requirements for where and how you create a file system snapshot. However, it is required that you **suspend LDT processes before you take the file system snapshot**. Suspending LDT ensures data and metadata consistency between files and LDT extended attributes.

You may choose to suspend LDT manually on the managed host using `voradmin ldt suspend` command or `fsfreeze -f`, or suspend LDT on the DSM.

Note: Be aware that suspending LDT on the DSM suspends LDT on the entire host.

After creating a file system snapshot, you can resume LDT processes on the GuardPoint using `voradmin ldt resume`, or `fsfreeze -u`, or resuming LDT on the DSM. Do not mix the use of `fsfreeze` and `voradmin ldt suspend` to pause and resume LDT. VTE suspends or resumes LDT processes during live transformation when freezing or unfreezing GuardPoint access using `fsfreeze -f` or `-u` option. Refer to the fsfreeze section in the *Agent Installation and Congifuration Guide* on the use of the fsfreeze command on a GuardPoint.

Note: You can make sure LDT is suspended at backup time by setting the host's QoS schedule.

You can mount a file system snapshot for data recovery. Configuration for GuardPoints must be duplicated over the mount point of the snapshot file system. Make sure to use the same LDT policy. Enabling GuardPoints over or under the snapshot mount point provides access to the protected files for recovery. You can choose to manually resume key rotation on the GuardPoints of the snapshot file system, although this is not necessary.

Following is an example of the fsfreeze command used to freeze access to the file system `/oxf-fs1` in order to create a snapshot of the file system device. This examples illustrates three GuardPoints enabled inside the file system namespace, `/oxf-fs1/gp-1`, `/oxf-fs1/gp-2`, and `/oxf-fs1/gp-3`. Executing the command `fsfreeze -f` targets any of the GuardPoints in the `/oxf-fs1` mount point and suspends LDT processes on all of the GuardPoints. Then it freezes access to the file system.

```
# fsfreeze -f /oxf-fs1/gp-1

# voradmin ldt list all
```

```
MDS_1:  type=file, nguards=1, name=/oxf-fs1/gp-3/::vorm:mds::
        Guard Table: version 1 nentries 1
        Guard 0: type=GP, state=REKEYING SUSPENDED (vadm), flags=GP LOCKED, gp=/
oxf-fs1/gp-3
                File List: count 4308

MDS_2:  type=file, nguards=1, name=/oxf-fs1/gp-2/::vorm:mds::
        Guard Table: version 1 nentries 1
        Guard 0: type=GP, state=REKEYING DIRTY, flags=GP LOCKED, gp=/oxf-fs1/gp-2
                File List: count 4308

MDS_3:  type=file, nguards=1, name=/oxf-fs1/gp-1/::vorm:mds::
        Guard Table: version 1 nentries 1
       Guard 0: type=GP, state=REKEYING SUSPENDED (vadm), flags=GP LOCKED, gp=/oxf-
fs1/gp-1
                File List: count 4308
```

After the file system snapshot is created, executing the `fsfreeze -u` command on any of the GuardPoints in the file system namespace unfreezes access to the file system and resumes LDT processes on all of the GuardPoints.

```
# fsfreeze -u /oxf-fs1/gp-1

# voradmin ldt list all
```

```
MDS_1:  type=file, nguards=1, name=/oxf-fs1/gp-3/::vorm:mds::
        Guard Table: version 1 nentries 1
        Guard 0: type=GP, state=REKEYING DIRTY, flags=GP LOCKED, gp=/oxf-fs1/gp-3
                File List: count 4308

MDS_2:  type=file, nguards=1, name=/oxf-fs1/gp-2/::vorm:mds::
        Guard Table: version 1 nentries 1
        Guard 0: type=GP, state=REKEYING DIRTY, flags=GP LOCKED, gp=/oxf-fs1/gp-2
                File List: count 4308

MDS_3:  type=file, nguards=1, name=/oxf-fs1/gp-1/::vorm:mds::
        Guard Table: version 1 nentries 1
        Guard 0: type=GP, state=REKEYING DIRTY, flags=GP LOCKED, gp=/oxf-fs1/gp-1
                File List: count 4308
```

Windows Backup and Snapshots

On Windows, most backup applications use Volume Snapshot Service (VSS) for the backup. Using VSS is required for backing up files in LDT GuardPoints. VSS service provides a consistent view of the data to backup and restore applications by taking a snapshot of the volume. Windows Backup uses the snapshot volume, while other applications can continue using the original volume for normal I/O operations. VSS snapshot volume uses a “Copy on write” mechanism to provide a consistent view of the data. Some of the high-level steps of the Windows backup process are:

1. Backup application takes a snapshot of the volume using the VSS service.
2. Backup application mounts this VSS volume to read the data to be backed up.
3. Vormetric agent uses the same policy as that of the original volume to protect these snapshots.
4. Vormetric agent applies the policy and rules to all the I/O requests coming from the backup and restore applications.

Restoring ESXi VM Snapshots of a Protected Host

Some organizations may use ESXi virtual machines for protected hosts and create VM snapshots to back up sensitive data. If you are restoring an older ESXi VM snapshot of a protected host, that contains an previous encryption key, follow these steps:

1. Restore the VM to update to the latest version of the encryption key.
2. In the DSM Management Console, choose **Hosts** > **Hosts**, then click the name of the VM host.
3. Click the **GuardPoint** tab.
4. Click **Re-Push Policies**.

LDT Backup and Restore Troubleshooting

Restored files to a GuardPoint protected with conflicting key rules

When restoring an encrypted file from backup media to an LDT protected GuardPoint without the Apply Key effect, and the file in the backup media does not have an LDT extended attribute, the file restored to the GuardPoint is set with an LDT extended attribute that specifies the current key version of the key in the policy associated with the data in the restored file. As the key and key version in the policy do not match the key that was applied to the data at the time of backup, the file restored to the LDT protected GuardPoint is unreadable.

When restoring an encrypted file from backup media to an LDT protected GuardPoint without Apply Key effect, and the key specified in the LDT extended attribute of the file in backup media conflicts with the key rules of the policy on the GuardPoint, the restore operation fails and flags the restored file in error. You can only remove the file, or truncate it, to clear the error status on the file. Access to such files, except remove or truncate, fail with an EINVAL error.

LDT Command-Line Administration: voradmin command

Use the `voradmin` utility to gather statistics and administer Live Data Transformation on a host. It has slightly different syntax and command capabilities depending on whether the host is running Linux or Windows.

Note: Refer to the voradmin man page for voradmin usage in Linux.

To use `voradmin`:

1. Log in to the host running the VTE agent with LDT enabled.
2. At the command line, type `voradmin`.
3. Follow the usage outputs onscreen to find the available commands and their syntax. The general syntax is:

```
# voradmin ldt <command> [args]
```

Command	Description
`voradmin ldt attr get|delete <guard path>|<object path>`	Get or delete attributes for a GuardPoint or Object path.
`voradmin ldt key <report|map <key> > <guard path>`	Creates a report, or map, containing statistics after each key rotation on a GuardPoint.
`voradmin ldt list all`	(Linux only) List all MDS files and GuardPoints undergoing live data transformation.
`voradmin ldt monitor [interval]`	(Windows only) Monitor LDT progress. If an interval is specified (in units of seconds), it continually updates the monitoring output at the specified interval.
`voradmin ldt rekey report <guard path> [<output file>]`	Generate a rekey report manually for a Guardpath.
`voradmin ldt resume <guard path>|all`	Resume guarding the GuardPoint directory.
`voradmin ldt space <guard path>`	Estimates disk space needed for metadata and mds store information.

Command	Description
`voradmin ldt stats [<guard path>] [<interval>]`	Obtains transformation LDT statistics such as: • Current rekey status • Start time • Estimated completion time • Percentage completed • Total data • Amount of data transformed • Total files • Number of files transformed • Number of files skipped • Number of files remaining for rekey

Upgrading or Downgrading Agent Software On an LDT Host

Consider the following when upgrading or downgrading software on protected hosts.

Upgrading

If a host contains an agent software version lower than 6.0, and you upgrade to version 6.0 or later, the LDT feature does not automatically enable on all upgraded hosts. You must:

1. Upgrade the DSM.
2. Install the LDT license.
3. Restart the 6.x agent, or reboot the host on which the Agent is installed.
4. Select the LDT option to enable it, once the LDT option displays in the host's menu.

After upgrading the DSM, the keys and policies on the DSM remain unchanged. They are standard keys and policies not usable with LDT. To use LDT, create versioned keys and policies set to type: Live Data Transformation.

On the DSM, select the Live Data Transformation option on the Edit > Host page to enable LDT on target host.

After upgrading a protected host to 6.x from an earlier version, existing policies, keys, and GuardPoints on the managed host remain unchanged and enforced.

Note: In v6.1, you can automatically register LDT with the DM installation.

For more information, see "Setting Up Live Data Transformation" on page 19.

Note: If you are using LDT with the Hadoop Distributed File System (HDFS), see the HDFS chapter in the VTE Agent Installation and Configuration guide for more information.

Downgrading

To roll back from 6.0 (or greater) to a version before 6.0, it is not sufficient to uninstall the 6.0 software and reinstall the earlier version. You must migrate GuardPoints protected with LDT encryption policies to a non-LDT policy first, reverse the LDT host's registration with the DSM, then re?register the host after installing the earlier software version. See "Removing LDT and Security Encryption" on page 78.

Moving an LDT GuardPoint from one LDT policy to another LDT policy

To change the LDT policy that an LDT GuardPoint uses, complete the following steps to ensure that the GuardPoint is migrated properly from one LDT policy to another LDT policy.

Scenario

The GuardPoint is currently attached to LDT-Policy-1, which rekeys from clear to LDT-Key-1. The objective is to migrate the data in the GuardPoint to another versioned key, LDT-Key-2. Migration to LDT-Key-2 requires detaching the GuardPoint from LDT-Policy-1, and then attaching it to LDT-Policy-2, assuming LDT-Key-2 is the versioned key specified in LDT-Policy-2. This is a two step migration:

1. Migrate a GuardPoint out of the LDT policy to protection under a standard policy

   The step requires cloning LDT-Key-1 to a non-versioned key, and migrating the GuardPoint out of the LDT policy. Because the cloned key and the the current version of LDT-Key-1 have the same key material, the data in the GuardPoint directory remains encrypted to the same key after migration of the GuardPoint out of LDT.
2. Migrate a GuardPoint protected under a standard policy to protection with an LDT policy.

   The step requires creating a new LDT policy, LDT-Policy-2, with the initial key set to the cloned key from the previous step, and the transformation key set to LDT-Key-2. Guarding and enabling the GuardPoint directory under the new LDT policy transforms the data in the GuardPoint to LDT-Key-2.

Note that you may set the transformation key to `clear_key`, instead of LDT-Key-2, in the second policy (LDT-Policy-2). The effect of `clear_key` is transformation of the data in the GuardPoint directory to clear text. Although the transformation key in LDT policies must be a versioned key, `clear_key` is an exception. It allows LDT GuardPoints to be transformed to clear using the LDT process.

To properly move the GuardPoint from LDT-Policy-1 to LDT-Policy-2:

Note: Ensure that no active processes or users are accessing files in the GuardPoint, or its subdirectories, until all steps have been completed.

1. Follow the instructions for, "Converting a GuardPoint from an LDT policy to a non-LDT policy" on page 78.
2. Assuming the current version of the LDT-Key-1 is version 5, then clone version 5 of the key to Cloned-LDT-Key_1v5.
   a. In the DSM Console, select **Keys > Agent Keys > Keys**.
   b. Click the name of the versioned key that you want to clone.
   c. Click **Clone** to clone the key.
3. Follow the instructions for,"Using voradmin To Estimate Disk Space Required for LDT (Linux)" on page 63.
4. Follow instructions for, "Converting a GuardPoint from an LDT policy to a non-LDT policy" on page 78 using the cloned key (`Cloned-LDT-Key-1v5`) as the Initial Key and `LDT-Key-2` as the key for the new LDT policy. The files are now rekeyed from `Cloned-LDT-Key-1v5` to `LDT-Key-2`.

Removing LDT and Security Encryption

If you want to stop using LDT on a GuardPoint or on a whole host, follow the instructions in this section.

Migrating a GuardPoint Out of LDT

Migrating a GuardPoint from LDT removes the security encryption. It also provides DSM administrators with the flexibility to relax the compliance requirement, when strict compliance for frequent key rotation on specific data is no longer mandatory. The following sections describe how to migrate a GuardPoint from LDT to a non-LDT policy, or to remove encryption protection from it.

Converting a GuardPoint from an LDT policy to a non-LDT policy

If you want to do more than just change the policy on a GuardPoint from an LDT policy to a non-LDT policy, see “Deleting LDT Metadata (Linux)” on page 81.

1. Clone the versioned key associated with the LDT GuardPoint to a non-versioned key.

   The clone function creates a new key with the same cryptographic encryption material as the current version of the cloned versioned key.

   This allows LDT to use the cloned key in a non-LDT policy to convert the GuardPoint from an LDT to a non-LDT managed policy.

   a. In the DSM Console, select **Keys > Agent Keys > Keys**.

   b. Click the name of the versioned key that you want to clone.

   c. Click **Clone** to clone the key.

   By default, the DSM uses the name of the versioned key, and the current version number of the versioned key, for naming the cloned key. You can choose a different name if desired.
2. Click **Hosts > <*host_name*> > GuardPoint tab**.
3. Select the GuardPoint and then click **Disable**.
4. After LDT disables the GuardPoint, click **Unguard** to delete the GuardPoint.
5. (Linux only) Ensure that the GuardPoint is removed on the managed host:

   ```
   # secfsd -status guard
   ```

   System Response:

   ```
   No GuardPoints configured
   ```
6. LDT creates extended attributes for every file under the GuardPoint as well as the GuardPoint directory. Now that the LDT policy does not manage the GuardPoint, you must remove the extended attributes for every file in the GuardPoint, type:

   ```
   # voradmin ldt attr delete /<guardPoint>
   ```
7. The command may take some time depending on the number of files in the GuardPoint.

Note: For all of the file system mount points that contain an LDT protected GuardPoint, you must clean up the metadata first. See “Deleting LDT Metadata (Linux)” on page 81.

8. Create a non-LDT policy using the cloned key created in step 1.

   a. Click **Policies > Manage Policies > Manage Policies**.

   b. Click **Add**.

c. In Policy Type, choose **Standard**.
d. In Key Selection Rules, click **Add**.
e. In Key, choose the cloned key from step 1.
f. Click **Select Key**.
g. Finish creating the policy as usual for a non-LDT policy.

9. Apply the non-LDT policy to the GuardPoint.

CAUTION
Make sure that you have removed all of the LDT metadata from the GuardPoint before applying the non-LDT policy.

a. Click **Hosts > <***host_name***> > GuardPoint tab**.
b. Click **Guard**.
c. In the *Guard File System* window, select the Policy that you just created.
d. Select *Type*: **Directory (Auto Guard)** or **(Manual Guard)**, and enter or browse to the directory to protect.

Remove Protection from a GuardPoint

When compliance may no longer require protecting data in a GuardPoint, you may choose to unprotect/decrypt it. Before removing protection from your GuardPoint, you must decrypt the data in your GuardPoint by setting it to clear. You have two options to decrypt your data:

- While a GuardPoint is protected and enabled under an LDT policy, you can use copy or backup/restore commands to save files in your GuardPoint to a location outside of your GuardPoint.
- Use the dataxform command to transform your GuardPoint to clear in an offline transformation process.

Copying Files to Decrypt Them

1. If you choose to copy your files, you must create a directory outside of the GuardPoint and then copy the files into the GuardPoint directory. After finishing copying, complete the following steps:In the DSM Management Console, click **Hosts** > **Hosts.**
2. Click the *hostName*.
3. Click the **GuardPoint** tab.
4. Click **Disable** to disable the GuardPoint.
5. After LDT disables the GuardPoint, click **Unguard** to delete the GuardPoint.
6. (Linux Only) Ensure that the GuardPoint is removed from the managed host. Type:

```
# secfsd -status guard
```

System Response

```
No GuardPoints configured
```

This completes removal of the GuardPoint under an LDT policy. You can now remove the original files and data within the GuardPoint namespace.

Using Dataxform Command to transform the files

If you choose to use the dataxform command to transform data in your GuardPoint to clear, use the `voradmin` command to verify that earlier versions of the versioned key are not in use on your GuardPoint.

Complete the following steps to clear all metadata in your GuardPoint. Then, transform your GuardPoint to clear.

1. Clone the key.
   a. In the Management Console, select **Keys > Agent Keys > Keys**.
   b. Click the name of the versioned key that you want to clone.
   c. Click **Clone** to clone the key.

      By default, the DSM uses the name of the versioned key, and the current version number of the versioned key, for naming the cloned key. You can choose a different name, if desired.
2. Create an offline dataxform policy to transform your GuardPoint from the cloned key created in step 1 to `clear_key`. See the "*Vormetric DSM Administrators Guide"* for more information.
3. In the DSM Management Console, click **Hosts > Hosts**.
4. Click the <*hostName*>.
5. Click the **GuardPoint** tab.
6. Click **Disable** to disable the GuardPoint.
7. After LDT disables the GuardPoint, click **Unguard** to delete the GuardPoint.

   Before you apply the offline data transformation policy to your GuardPoint, you must clean up the LDT metadata from your GuardPoint. LDT creates extended attributes for every file under the GuardPoint, as well as the GuardPoint directory. Now that the LDT policy does not manage the GuardPoint, you can remove the extended attributes for every file in the GuardPoint.
8. Remove the extended attributes of files in a GuardPoint, type:

   ```
   # voradmin ldt attr delete <guardPoint>
   ```

   The command may take some time depending on the number of files in the GuardPoint. After metadata deletion is complete, you can apply the offline transformation policy on the GuardPoint.
9. On the DSM, guard and enable the GuardPoint with the offline dataxform policy created in step 2.
10. After enabling your GuardPoint, run the dataxform command on the managed host to transform the GuardPoint to a `clear_key`, type:

    ```
    # dataxform --rekey --gp /<guardPoint>/ --preserve_modified_time --
    preserve_access_time --cleanup_on_success
    ```

11. After completion of dataxform, unguard the GuardPoint.
    a. Click **Hosts > <***host_name***> > GuardPoint** tab.
    b. Select the GuardPoint and click **Disable**.
    c. After LDT disables the GuardPoint, click **Unguard** to delete the GuardPoint.
12. Remove the GuardPoint from the dataxform policy on the DSM.
    a. Click **Policies** > <**policyName**>.
    b. Select the **GuardPoint**.
    c. Click **Delete**.
    d. Click **Apply**.

Deleting LDT Metadata (Linux)

To remove the metadata associated with the GuardPoint, you must run the voradmin command on the managed host to remove the LDT metadata associated with the GuardPoint.

WARNING

Before you attempt to remove an MDS file, make sure that no LDT-protected GuardPoints remain configured under the file system mount point.

In the following example:
`/oxf-fs1` is the mount point
`/oxf-fs1/gp1` is the LDT-protected GuardPoint

To delete the metadata associated with the GuardPoint:

1. Ensure that the GuardPoint is not enabled on the host. Run the command below and verify that the GuardPoint pathname does not appear in the output of the secfsd command, type:

   ```
   # secfsd -status guard
   ```

2. Run the voradmin command to remove LDT attributes on the GuardPoint, type:

   ```
   # voradmin ldt attr delete <guardpoint path>
   ```

   Example:

   ```
   # voradmin ldt attr delete /oxf-fs1/gp1
   ```

   System Response:

   ```
   LDT metadata has been removed from all files in guardpoint /oxf-fs1/gp1
   ```

If the last transformation on your GuardPoint completed when the VTE agent was installed at v6.0.2, you may have to remove the MDS file that is located in the mount point directory of the file system in which GuardPoint resides. Before you remove the MDS file, make sure that the MDS file is not in use with other GuardPoints under the same file system mount point. For example, if you removed the metadata for GuardPoint `/oxf-fs1/gp-1` where `/oxf-fs1` is the mount point of the mounted file system, make sure that `/oxf-fs1` does not appear in the output of the `secfsd -status guard` command, and then proceed:

1. Remove the MDS file from the mount points or GuardPoints. As of the VTE 6.1 release, LDT always creates and manages the MDS file inside the GuardPoint directory. Earlier versions created and managed the MDS file inside mount point directory while sharing the MDS file with all the GuardPoints under the mount point. Type:

   ```
   # voradmin ldt rmstore <mount_point>
   ```

   Example:

   ```
   # voradmin ldt rmstore /oxf-fs1
   ```

   System Response:

   ```
   Enter YES if /oxf-fs1 does not include any GuardPoints associated with an
   LDT policy ->YES
   MDS file /oxf-fs1/::vorm:mds:: has been removed.
   ```

2. Verify that the metadata store has been removed from the secfs mount points, type:

   ```
   # ls -altr <file_system_mount_point>
   ```

Example:

```
# ls -altr /oxf-fs1
```

You should not see `/oxf-fs1/::vorm:mds::` listed.

Deleting Metadata (Windows only)

Use the following command to delete the VTE extended attribute for a file or a GuardPoint. This is useful when removing a GuardPoint from under an LDT policy (see “Migrating a GuardPoint Out of LDT” on page 78).

```
# voradmin ldt attr delete [<file name path> | <guard path>]
```

For an overview of `voradmin ldt`, see “LDT Command-Line Administration: voradmin command” on page 75.

Removing LDT from a Host

Once you have registered a host and enabled LDT, you cannot disable the LDT feature by unchecking the LDT box. You must unregister the host from the DSM, then register it again without LDT. When you remove the LDT feature from a host entirely, the host’s LDT license becomes available for use on another host.

1. Stop all applications from accessing data in LDT GuardPoints on the host.
2. Migrate data in every LDT GuardPoint using the steps described in the section “Remove Protection from a GuardPoint” on page 79.

WARNING

Potential data loss. Ensure that you have decrypted the data and, optionally, copied it out of the GuardPoint. Once the agent software is removed, access to data is no longer controlled by VTE. If the data was encrypted, it remains encrypted, and there is no way to read it.

3. Remove the GuardPoints on the host from the DSM.
   a. Remove the LDT metadata from those GuardPoints.
   b. Remove the MDS files associated with those GuardPoints, if necessary. See “Deleting LDT Metadata (Linux)” on page 81 for more information.
4. Remove the host from the DSM. Follow the steps in “Uninstalling Agents” in the *Vormetric Transparent Encryption Agent Installation and Configuration Guide*.
5. Re-install the agent on the host. Again, use the *Vormetric Transparent Encryption Agent Installation and Configuration Guide*.
6. Register the host with the DSM. This time, do not select the Live Data Transformation option. See “Registering an LDT Host” on page 22.

Uninstalling the Agent while LDT is Rekeying GuardPoints

You cannot uninstall an agent if LDT is rekeying GuardPoint(s) on that agent.

If you want to uninstall an agent from a host, which contains LDT GuardPoints, you must first unencrypt the files on each LDT GuardPoint using the steps noted in the previous sections.

Chapter 6: Troubleshooting LDT

The LDT administrator needs to monitor and respond to runtime statistics and alerts. This chapter contains the following sections:

Monitoring and Statistics

This section describes how to find the runtime status of LDT and how to get statistics about its operation.

Note: When you check GuardPoint status on the DSM, be aware that the status is not relayed in real time. There can be a delay of several minutes before the DSM becomes aware of events on the host. When you configure the host and DSM for one-way communication, the delay is longer, up to an hour, because LDT sends statistics to the DSM only once per hour.

To find out when the DSM last received a status update from the host, check the Last Status Update timestamp in the GuardPoint Status dialog (for information about how to display this dialog, see "Obtaining Statistics in the UI with GuardPoint Status" on page 83). For the most up-to-date statistics, inspect the host. Run the voradmin command on the managed hosts to retrieve status and statistics pertaining to GuardPoints.

To find out whether your host is configured for one-way communication, open the DSM Management Console, choose **Hosts > Hosts**, click the name of the host, and look for the FS Agent One Way Communication checkbox.

Obtaining Statistics in the UI with GuardPoint Status

During and after data transformation, you can view statistics about any GuardPoint on any host in your system. Either use the DSM Management Console browser UI, as described in this section, or use the `voradmin` command, as described in "Obtaining LDT Statistics at the Command Line" on page 85.

To display the GuardPoint status in the Management Console UI:

1. In the DSM Management Console, choose **Hosts** > **Hosts**.
2. Click the name of the host you want to inspect.
3. Click the **GuardPoint** tab.
4. In the Status column, click the icon.

The GuardPoint Status window opens. The window displays the following values:

GuardPoint Status Field	Description
Reason	If the key rotation was initiated manually on an LDT host, the requester selected a reason such as “normal maintenance” or “key compromised.”
GuardPoint Status	Status of the GuardPoint.
GuardPoint State	State of the GuardPoint.
Usage	Displays the reference count for the GuardPoint.
Guarded on	Date when the policy went into effect on this GuardPoint.
Policy Name	Name of the policy.
Policy Version	The current policy version. If you make any changes to a policy after creating it, the policy version number increases by 1. The first version is number 0.
Last Status Update	Time stamp that displays when the DSM last received status information from the agent on the protected host. There can be a delay between status updates from agent to DSM. If it is imperative that you see the most recent data, go to the agent and use the voradmin command to inspect it. For an overview of voradmin ldt, see “LDT Command-Line Administration: voradmin command” on page 75.
Policy Aggregate Key Version	Indicates the version of the key used to transform data.
Transformation Status	A one-word summary of the current (or most recent) Live Data Transformation. The same value is shown in the Rekey Status column on the host’s GuardPoint tab. For details about each status code, see “Checking the Rekey Status” on page 42.
Last Transformation Completion Time	Timestamp from the last time that transformation completed successfully.
Last Transformation Start Time	Timestamp showing the last time transformation was started. If this time is later than the Last Transformation Completion Time, the transformation did not complete normally. The Transformation Status should reflect this, such as with Incomplete.
Estimated Rekey Completion Time	If transformation is underway, this field displays the rekey duration.
Total Number of Files Deleted	Number of files in the GuardPoint that were deleted by users while data transformation was underway.
Total Files to Be Transformed	Number of files found during the initial scan of the GuardPoint. For more information, see “LDT Runtime Flow” on page 14.
Total Bytes to Be Transformed	Size of all of the files found during the initial scan of the GuardPoint.
Total Bytes Transformed	Number of bytes in the GuardPoint that were successfully transformed in the most recent LDT operation. If this does not match Total Bytes to Be Transformed, look for Total Files Skipped or Total Files Errored.

GuardPoint Status Field	Description
Total Files Transformed	Number of files in the GuardPoint that were successfully transformed in the most recent LDT operation. If this does not match Total Files to Be Transformed, look for Total Files Skipped or Total Files Errored.
Total Files Skipped	Total number of files that LDT did not transform in this pass. The files were intended for transformation, but were not transformed for some reason. For example, (on Windows) the file is read-only, or it is an executable file that is currently running. On a Windows host, transformation continues on the next available file in the GuardPoint, and LDT maintains a list of the files that it skipped.
Total Files Errored	Total number of files that triggered alerts. See "Alerts Playbook" on page 88.

Obtaining LDT Statistics at the Command Line

You can obtain LDT statistics from the command line using the following command:

```
# voradmin ldt stats <guard path> [interval]
```

This command displays transformation statistics for a GuardPoint, or for all GuardPoints, if none are specified. If you specify an interval (in units of seconds), the command continually updates the statistics on the given interval. If no GuardPoint is specified, the command returns the aggregate statistics at the host level, including specific statistics related to QoS. If a GuardPoint is specified, statistics include:

- Current rekey status
- Rekey Start time
- Last rekey completion time
- Estimated Rekey completion time
- Data Statistics including:
  - Amount of data transformed
- File statistics including:
  - Number of files transformed
  - Number of files skipped
  - Number of files remaining for rekey

Obtaining a Rekey Report

LDT generates a report automatically after completion of a key rotation. The report lists the files in the GuardPoint and the key and version of the key applied to each file. This kind of report may be a common compliance requirement.

The administrator can also request a rekey report during rekey, if the need to see the partial transformation results so far.

About the rekey report

On Linux hosts, LDT writes rekey reports to a local file on the agent host in `/var/log/vormetric/`. The file name begins with `ldaudit-log-`. It is followed by the file system directory name, GuardPoint directory name, and a timestamp. For example, for a GuardPoint at `/oxf-fs2/gp2` with rekey completed on November 19, 2016, just after 2:00 p.m. (hour 14), the rekey report file name would be `ldtaudit-log-_oxf-fs2_gp2-2016111914917`.

The report includes:

- Total number of files in the GuardPoint
- Number of files transformed
- Rekey start and end times
- List of all files transformed
- Applied key and key version for each file (for example, files in different resource sets might have used different keys)

Manually generating a rekey report

To generate a rekey report manually, use the following command:

```
# voradmin ldt rekey report <GuardPoint> [<logfile>]
```

In `<GuardPoint>`, type the GuardPoint path. In `<logfile>`, you can optionally direct the output to a file. If no logfile is specified, the report displays on `stdout`.

Monitoring Ongoing LDT Operations at the Command Line (Windows only)

Use the following syntax to monitor LDT progress. If an interval is specified (in units of seconds), it continually updates the monitoring output at the specified interval.

```
# voradmin ldt monitor [interval]
```

For an overview of `voradmin ldt`, see “LDT Command-Line Administration: voradmin command” on page 75.

Protecting LDT GuardPoints against Failure in Underlying Filesystems (Linux)

LDT Recovery Challenges

The main challenge with LDT recovery is a failure to access files or specific blocks in files inside a GuardPoint. Before LDT runs the recovery process on a GuardPoint, the underlying file system was recovered before file system was mounted. File system recovery may create orphan files in the `lost+found` directory inside the mount point. If an orphan file belongs to the GuardPoint, LDT cannot recover the file as the file is no longer in the same directory as it was prior to the crash while it was undergoing rekey.

LDT Recovery Enhancement

LDT has been improved for handling inconsistencies in the underlying file system during execution of LDT recovery. When enabling a GuardPoint after a system crash, LDT performs consistency checks on the files undergoing rekey at the time of crash before enabling GuardPoint. If the LDT recovery process is unable to recover any of the affected files, the GuardPoint will not be enabled and LDT will send the following alert to the DSM:

```
[CGS3266E] LDT-ALERT: Cannot enable guard point [GuardPoint] due to inconsistencies
in underlying file system encountered during LDT recovery.
```

You can check the status of a GuardPoint and the reason for failing to enable a GuardPoint using the `secfsd -status guard` command. For example, the GuardPoint `/mnt/gp` failed to guard because `Guard point needs LDT recovery` as reported in the `Reason` column for the following example GuardPoint.

```
GuardPoint Policy Type   ConfigState Status      Reason
---------- ------ ----   ----------- ------      ------
/mnt/gp    LDT220 manual guarded     not guarded Guard point needs LDT recovery
```

Details on the specific issues encountered, including any files that could not be recovered, are reported to a log file in the `/var/log/vormetric` directory. The log file name is in the format:

```
ldt_recovery_log:<guardpoint_name>:<timestamp>.txt
```

The log file name starts with `ldt_recovery_log`, followed by the GuardPoint pathname and the date and time of the recovery attempt. The GuardPoint pathname and date and time are separated with "`:`". For example:

```
/var/log/vormetric/ldt_recovery_log:_mnt_gp:2018-12-08-14:6:12.txt
```

Refer to the issues listed in the log file, and resolve those issues before enabling the GuardPoint again. A GuardPoint cannot be enabled in subsequent guard attempts until those issues are resolved. Each attempt generates a new log file. Check the LDT log file for missing files whose inode numbers match inode numbers of orphan files in `lost+found`. If there is a match, restore the file from `lost+found` to the pathname specified in the LDT recovery log file. Be sure to run the `mv` command to restore the file to the original location, do not run the `cp` command. After resolving all or some of the reported issues, you must run the command `voradmin ldt recover <GuardPoint>` to repeat the recovery process.

```
voradmin ldt recover <GuardPoint>
```

Running this command resolves the problems that can be corrected and clears the failed recovery status on the GuardPoint, allowing the GuardPoint to enable automatically within 30 seconds. If the GuardPoint does not enable, you can enable it using `secfsd -guard` command, if manual guard, or enable it on the DSM, if auto-guard. Note that if you are unable to resolve all of the reported issues, you accept the loss of some data or files, as reported in the latest recovery log file for the GuardPoint, when you run the command `voradmin ldt recover <GuardPoint>`.

Alerts

LDT sends two alerts to DSM if LDT fails to resolve the issues encountered during recovery.

The alert below is an error report sent to DSM whenever LDT encounters failure during recovery prior to enabling GuardPoint. This alert also reports the GuardPoint specified in the message is not enabled:

```
[CGS3266E]  LDT-ALERT: Cannot enable guard point [GuardPoint] due to
inconsistencies in underlying file system encountered during LDT recovery.
```

The alert below is a warning report sent to DSM only if “voradmin ldt recover” encounters errors and continues anyway. This alert reports the GuardPoint specified in the message has been enabled with some files in error status.

```
[CGS3267W]  LDT-ALERT: LDT manual recovery on guard point [GuardPoint] completed
with [#] errors.
```

The full message body does not appear in the logs on DSM until a new release of DSM in future. In the meantime, the DSM shows the message ID without the message body.

Alerts Playbook

This section lists the LDT alerts and describes what to do in each case. LDT generates alerts when issues arise that require attention from the DSM administrator. Without prompt attention, it can delay the rekey process or cause the process to remain incomplete.

Failure to Suspend or Resume LDT Operation

LDT-ALERT: Failed to suspend rekey on GuardPoint [GuardPoint]

Solution

An I/O error is the most common cause of failure when updating the persistent state of a GuardPoint. For I/O errors, you must fix the problem at the host OS or storage level.

The appropriate corrective action depends on when this problem occurs and the reason for the suspension. If the suspend request was at the host level as part of a QoS schedule, or it was initiated by a user on the DSM, the failure occurred at the host level rather than the specified GuardPoint.

Otherwise, the source is probably the initiation of a suspension on a specific GuardPoint before backup. If the backup operation is in progress when this alert message occurs, you must fix the cause of the suspension failure and then restart the backup. If the backup has already completed when the alert message occurs, the backup image on the GuardPoint may have inconsistent data and LDT metadata. Discard this backup image and do a fresh backup.

If you cannot find and fix the host OS or storage issue, contact Vormetric Support for troubleshooting and recovery.

LDT-ALERT: Failed to resume rekey on GuardPoint [GuardPoint]

LDT-ALERT: Failed to resume rekey on all GuardPoints

Solution

An I/O error is the most common cause of failure when updating persistent state of a GuardPoint. For I/O errors, you must fix the problem at the host OS or storage level.

The appropriate corrective action depends on when this problem occurs and the source of the problem. If the resume request was at the host level as part of a QoS schedule, or the resume request was initiated by a user on the DSM, the failure occurred at the host level rather than the specified GuardPoint.

Otherwise, the source is probably initiation of a resume operation on a specific GuardPoint after a backup is completed.

If you cannot find and fix the host OS or storage issue, contact Vormetric Support for troubleshooting and recovery.

Insufficient Resources

The following alerts trigger when there are not enough resources for LDT operations.

LDT-ALERT: Aborting rekey of file [FileName] due to insufficient disk space

The key rotation process was stopped on a GuardPoint because there was not enough available disk space.

Solution

Resize or free up space in the file system where the PathName resides. Key rotation automatically starts again when sufficient free space becomes available.

LDT-ALERT: Failed to launch transformation on [GuardPoint]

LDT failed to launch the data transformation process on the specified GuardPoint.

Solution

The most common causes of this failure are:

- Low amount of system resources, such as free disk space.
- Check the system logs for insufficient free space in underlying file systems. You can free up space by removing older files or resizing your underlying file system. You may resize the file system on-line or off-line.

For other issues, contact Customer Support.

LDT-ALERT: QoS failed to start

The Quality of Service subsystem failed to launch QoS services due to lack of system resources.

Solution

A low amount of available memory is the most common cause of this failure. Contact Vormetric Customer Support.

LDT-ALERT: Skipped key rotation on GuardPoint [GuardPoint] due to insufficient disk space

LDT-ALERT: MDS file [PathName] exceeded diskspace quota on gp [GuardPoint]

There was not enough storage space in one or more file systems containing GuardPoints undergoing rekey.

Solution

Check the file systems where your GuardPoints reside to see how much space it is using. Add more storage space as needed. After the condition is corrected and disk space is available, LDT operations resume automatically.

LDT-ALERT: Low space on guard point [GuardPoint], increase free space or LDT will be suspended.

Available storage space in the file system containing the GuardPoint is nearing the threshold below which LDT rekeying cannot continue. If available space does drop below that threshold, rekeying will be automatically suspended.

Solution

Check the file system where your GuardPoint resides to see how much space is available. Make more free space available if necessary. After the condition is corrected and storage space is available, rekeying resumes automatically.

Failed to Update LDT Attribute

The following alerts trigger when LDT extended attributes are not properly updated.

LDT-ALERT: Failed to update LDT attribute on GuardPoint [GuardPoint]. Error: [ErrorNumber]

Solution

I/O error is the most common cause of failure when updating LDT extended attributes. For I/O errors, fix the problems at the host OS or storage level. In Linux, if you experience an I/O error, identify the file that had the error, and restore it from backup. See “Backing Up and Restoring LDT GuardPoints” on page 65.

In Windows, you can recover a file that has corrupt metadata as follows:

- If the file is fully encrypted, remove the metadata. Type:

  ```
  # voradmin ldt attr delete <file name path>
  ```

  Then apply the same policy that was used to encrypt the data.
- If the file was only partially rekeyed, restore the file from a backup.

If you cannot find and fix the underlying host OS or storage issue that is causing corrupt metadata, contact Vormetric Support for troubleshooting and recovery.

LDT-ALERT: Failed to update LDT attribute

LDT failed to create or update an LDT extended attribute of a file in a GuardPoint.

Solution

An I/O error is the most common cause of failure when updating an LDT extended attribute. For I/O errors, fix the problem at the host OS or storage level.

If you cannot find and fix the host OS or storage issue, contact Vormetric Support for troubleshooting and recovery.

Rekey Stopped

The following alert is specific to Windows and is triggered when rekey is stopped before it completes normally.

LDT-Alert: Suspending rekey of binary file [PathName] during its execution

LDT operations encountered an executable file that is running.

LDT cannot rekey a running executable file until the execution of the binary file stops.

Solution (Windows)

1. Obtain the name of the executable file from the LDT-ALERT message.
2. Stop execution of the binary file.

Incomplete Key Rotation

LDT-ALERT: Failed to rotate key on GuardPoint [GuardPoint]. Another rekey operation is already in progress

LDT-ALERT: Key rotation failed on GuardPoint [GuardPoint]

LDT failed to complete the key rotation process on the specified GuardPoint. This may occur if a key rotation process is already in progress on the GuardPoint. This pre-existing rekey operation could be active, or it could be in a suspended state, either because of the QoS schedule or a manual pause initiated by the administrator. See “Suspending and Resuming Rekey and/or Scan Phase” on page 44.

Solution

1. On Linux, you can run the following command to see whether key rotation is in progress on the GuardPoint, or check the rekey status of GuardPoints as outlined in step 2 below:

   ```
   # voradmin ldt list all
   ```

2. On Linux or Windows, check the rekey status of GuardPoints in the Management Console UI.
3. If key rotation is already in progress on a GuardPoint, wait for the key rotation to complete.
4. On the GuardPoint page on the DSM, press **Re-push Policies** to manually re-push the policies to the host to initiate key rotation on GuardPoints ready for key rotation.

   Contact Vormetric Support if the cause of key rotation failure is not a key rotation in progress.

Skipped Key Rotation

The following alerts trigger when LDT skips key rotation.

LDT-ALERT: Skipped key rotation on GuardPoint [GuardPoint]. It is on a read-only file system

Key rotation is skipped on a specified GuardPoint because the mount point where the GuardPoint resides does not permit write operations.

Solution

1. Remount the file system where the GuardPoint resides, and change the mount option to read/write.
2. On the GuardPoint page on the DSM, press **Re-push Policies** to manually re-push the policies to the host to initiate key rotation on GuardPoints ready for key rotation.
   LDT-ALERT: Failed to rotate key on GuardPoint [GuardPoint] during pre-commit

LDT-ALERT: Failed to rotate key on GuardPoint [GuardPoint] during commit

LDT-ALERT: Skipped key rotation on GuardPoint [GuardPoint]. Error: [ErrorNumber]

```
LDT-ALERT: Failed to update LDT attribute on GuardPoint [GuardPoint]. Error:
[ErrorNumber]
```

LDT failed to start a key rotation process on a GuardPoint during a guard operation or when processing a key rotation notification from the DSM. For [*ErrorNumber*], a Linux error number is substituted, such as `errorcode 17`.

Solution

The host returns error code 17 during LDT key rotation if it cannot perform the key rotation because there is already a rekey in progress. This pre-existing rekey operation could be active, or it could be in a

suspended state, either because of the QoS schedule or a manual pause initiated by the administrator. See “Suspending and Resuming Rekey and/or Scan Phase” on page 44.

An I/O error is the most common cause of failure when updating the persistent state of a GuardPoint. For I/O errors, fix the problem at the host OS or storage level.

If you cannot find and fix the host OS or storage issue, contact Vormetric Support for troubleshooting and recovery.

Failed to Update LDT Metadata During Scan Phase

LDT triggers the following alerts when LDT cannot properly update metadata during the scan phase. (See “LDT Runtime Flow” on page 14 for information about the scan phase.)

LDT-ALERT: Scan error on [*GuardPoint*] removing MDS guard for relaunching dataxform

LDT failed to update metadata when restarting a scan on the specified GuardPoint.

Solution

An I/O error is the most common cause of failure when updating LDT metadata. For I/O errors, fix the problem at the host OS or storage level and then restart operations.

If you cannot find and fix the underlying host OS or storage issue that is causing the error, contact Vormetric Support for troubleshooting and recovery.

LDT-ALERT: Online Dataxfrom failed during post scan stage on [*GuardPoint*]

LDT failed to update metadata when transitioning from scan to rekey phase on the specified GuardPoint.

Solution

An I/O error is the most common cause of failure when updating LDT metadata. For I/O errors, the problem must be fixed at the host OS or storage level and the operation restarted.

If you cannot find and fix the underlying host OS or storage issue that is causing the error, contact Vormetric Support for troubleshooting and recovery.

File system inconsistencies after system crash

LDT sends two alerts to DSM if LDT fails to resolve the issues encountered during recovery.

The alert below is an error report sent to DSM whenever LDT encounters failure during recovery prior to enabling GuardPoint. This alert also reports the GuardPoint specified in the message is not enabled:

LDT-ALERT: Cannot enable guard point [GuardPoint] due to inconsistencies in underlying file system encountered during LDT recovery

The alert below is a warning report sent to DSM only if “voradmin ldt recover” encounters errors and continues anyway. This alert reports the GuardPoint specified in the message has been enabled with some files in error status.

LDT-ALERT: LDT manual recovery on guard point [GuardPoint] completed with [#] errors.

Solution

See “Protecting LDT GuardPoints against Failure in Underlying Filesystems (Linux)” on page 86.

Error Messages

This section describes runtime error messages. For information about other types of runtime messages, see "Alerts Playbook" on page 88 and "Warning and Info Messages" on page 95.

Failed to Transform File During Rekey

LDT: Rekey failed for file [PathName] on GuardPoint [*GuardPoint*]

LDT: Extended attribute of inode [InodeNumber] is corrupted under GuardPoint [*GuardPoint*]

LDT could not complete transformation on a file.

Solution

An I/O error is the most common cause of failure when updating LDT metadata. For I/O errors, fix the problem at the host OS or storage level, and then restore the file from a backup.

If you cannot find and fix the underlying host OS or storage issue that is causing the error, contact Vormetric Support for troubleshooting and recovery.

Failure to Suspend Live Data Transformation

LDT: Failed to suspend rekey on all GuardPoints

LDT operations could not be suspended on the host.

Solution

An I/O error is the most common cause of failure when updating the persistent state of a GuardPoint. For I/O errors, fix the problem at the host OS or storage level.

The suspend request was at the host level as part of a QoS schedule, or the suspend request was initiated by a user on the DSM.

If a backup operation is in progress when this message occurs, you must fix the cause of the suspend failure and then restart the backup. If the backup has already completed when the alert message occurs, the backup image on the GuardPoint may have inconsistent data and LDT metadata. Discard this backup image and do a fresh backup.

If you cannot find and fix the host OS or storage issue, contact Vormetric Support for troubleshooting and recovery.

Failure to Start or Stop Transformation

The following general messages are recorded when there are errors attempting to start or stop LDT. Examine system logs for additional information as to the cause. If you cannot find and fix the underlying host OS or storage issue that is causing the error, contact Vormetric Support for troubleshooting and recovery.

LDT: Failed to abort key rotation on GuardPoint [*GuardPoint*]

LDT: Failed to start

LDT: Failed to stop

LDT: Failed to exit

Failure to Restart Transformation

The following message is recorded when an attempt to restart transformation after a system reboot fails because the file system is mounted as read-only. Transformation on the specified GuardPoint cannot continue until the file system is mounted with write permission.

LDT: Skipped LDT recovery on read-only file system [*GuardPoint*]

Temporary Failure to Start Transformation on a File

The following messages are recorded when there are communication issues or lack of resources on the host. Once the condition is corrected, transformation of the file continues automatically.

LDT: Insufficient memory condition encountered during LDT on GuardPoint [*GuardPoint*]

LDT: Encryption key for file [*PathName*] unavailable for LDT, possibly due to loss of communication to DSM

LDT: Aborting rekey of file [*PathName*] due to lack of free memory. Try closing other application to resolve the issue

Transient Condition while enabling GuardPoint

The following message is recorded when subsequent attempts to guard a directory is attempted while the LDT GuardPoint initialization is already in progress. because the GuardPoint initialization is already in progress. During the GuardPoint initialization, the system initializes MDS file associated with the GuardPoint in preparation for LDT. Initialization of theMDS file can take a few minutes. During this time, if the system retries the guard operation, the following message displays. Once the initialization completes, the ongoing guard operation can then succeed.

Not re-guarding path [path] (Reason: Guard point initialization already in progress)

Not re-guarding path [path] from container [container] (Reason: Guard point initialization already in progress)

Failed to transform passthrough files for AD database files (Windows Only)

LDT skipped the transformation of passthrough files for all AD database files. The problem occurs when the AD database remains in the default folder location. To fix, move the AD database to any folder other than "c:\windows" or "c:\program files."

Skipping transformation of passthrough file [%s]. The file resides in the boot directory. VTE cannot encrypt boot directory files.

Skipping transformation of passthrough file [%s]. VTE cannot encrypt these files. They are already encrypted using NTFS encryption or compressed.

Skipping transformation of passthrough file [%s]. File is not in a GuardPoint directory.

Warning and Info Messages

This section describes warning and informational messages. For other types of runtime messages, see “Alerts Playbook” on page 88 and “Error Messages” on page 93.

Stopping Transformation of a File on Volume Dismount (Windows only)

The following warning message is recorded when transformation of a file is stopped because the containing volume is dismounted.

LDT: Volume dismounted. Aborting transformation at file [*PathName*]

Issues with Policy or System Configuration

The following warning messages are recorded when VTE cannot perform LDT on the GuardPoint, because there are errors in the policy associated with the GuardPoint, or the file system containing the GuardPoint is not supported by LDT. Files in the GuardPoint can still be accessed, but no Live Data Transformation occurs.

LDT: The GuardPoint [*GuardPoint*] does not have a valid transformation policy, there is no new key rule

LDT: The GuardPoint [*GuardPoint*] does not have a valid transformation policy, there is no key_op rule

LDT: The GuardPoint [*GuardPoint*] does not support online rekey, only file operations from the offline data transform process will be allowed

Failure to Enable GuardPoint During Cleanup

LDT records the following informational message when a user attempts to enable a newly added GuardPoint. This message displays if the GuardPoint directory was previously guarded with an LDT policy and the LDT metadata cleanup is in progress when user is guarding under the new policy. Retry the operation once the cleanup completes.

LDT: Cannot enable GuardPoint [*%s*] during LDT clean-up process

General LDT Operations

The following informational messages are recorded during various LDT operations. No action is required.

LDT: Successfully suspended rekey on GuardPoint [*GuardPoint*]

LDT: Successfully suspended rekey on all GuardPoints

LDT: Successfully resumed rekey on GuardPoint [*GuardPoint*]

LDT: Successfully resumed rekey on all GuardPoints

LDT: Rekey operation completed on GuardPoint [*GuardPoint*]

Missing LDT extended attribute

LDT: Extended attribute of inode [InodeNumber] is missing under GuardPoint [GuardPoint]

This warning message reports that the file with the specified inode number in the specified GuardPoint directory does not have an LDT extended attribute, therefore, access to the file is denied.

Solution:

LDT cannot determine the encryption key associated with the data in the file, therefore, you can only remove the file.

Locking Contention

LDT: Exclusive access for rekey delayed on inode [InodeNumber].

LDT: Exclusive access for rekey granted after delay on inode [InodeNumber].

The following messages are recorded during the rekey process on a file. When user access to the file is very high, it causes a high degree of locking contention between the rekey process and user access. The second message reports when the contention is no longer in effect and the rekey process has resumed accessing file to rekey.

LDT: Exclusive access for rekey delayed on inode [InodeNumber]

LDT: Exclusive access for rekey granted after delay on inode [InodeNumber]

Initiation and completion of LDT metadata cleanup

The following messages are recorded at the beginning and completion of LDT metadata cleanup through voradmin command.

LDT: Metadata will start getting removed from all files in GuardPoint [GuardPoint]

LDT: Metadata has been removed from all files in GuardPoint [GuardPoint]

Upgrading VTE Agent

On Windows, you cannot upgrade the VTE agent if there is an LDT GuardPoint with an active rekey in progress, that is at the mount point. On Linux, LDT is forced-suspended when upgrading VTE.

Reboot is required when upgrading earlier versions of VTE Agent to 6.0.1, if LDT is in-progress on a GuardPoint. Perform following steps to upgrade to v6.0.1:

1. From the DSM GUI, click **Host > Hosts >** *hostName* **> GuardPoint tab.**
2. Click **Suspend Rekey**.
3. Click **Refresh** and ensure that the GuardPoint(s) being rekeyed now show a status of Suspended.
4. Reboot the host.
5. Run the 6.0.1 VTE agent install program and upgrade the agent. Allow the upgrade to complete.
6. From the DSM GUI, click **Host > Hosts >** *hostName* **> GuardPoint tab**.
7. Click **Resume Rekey**.
8. Click **Refresh** and ensure that the GuardPoint(s) being rekeyed now show a status of Rekeying.

Advisory

The following information provides guidance for a better user experience.

All Platforms

Binary Re-signing

Any executable that is part of either a Signature set or a Host setting, and that resides in a GuardPoint that uses an LDT policy, will use different signatures for an LDT key rotation. The result is that the Host Settings binaries will no longer be authenticated, or that the Signature Set policy rules will no longer trigger for those binaries.

To prevent these issues, the Security Administrator must manually re-sign each affected binary after each key rotation.

Alternatively, VTE for Linux 6.1.2 and VTE for Windows 6.2.0 can generate unencrypted signatures of binaries inside GuardPoints to avoid these problems. See the section titled “Re-Signing executable files on secfs GuardPoints” in the “Special Cases for VTE Policies” chapter of the VTE Agent Installation and Configuration Guide.

Check for available disk space for LDT metadata

Before launching LDT on a GuardPoint:

1. Check the available free disk space in the file system where your GuardPoint resides.
2. Type the following command to check the disk space requirement of LDT on a target GuardPoint:

   Example:

   ```
   # voradmin ldt space /oxf-fs1/gp1
   ```

System Response:

```
/oxf-fs1/gp1: found 1501 files without LDT extended attributes

LDT disk space requirements: total 169MB (LDT attributes=6MB, MDS=163MB)
```

As reported in the example output, LDT requires 169MB of available disk space to launch and execute live data transformation on `/oxf-fs1/gp1`.

Note: Make sure that the free space in your file system exceeds the disk space requirement for LDT.

LDT Requirements for Backup

VTE offers the option to backup encrypted data from files through a security rule, which skips the Apply Key function when reading encrypted files by backup applications or processes. Such files can also be restored from backup streams, to the same or a different GuardPoint, without Apply Key. If you choose to restore to another GuardPoint, the target GuardPoint must be protected with the same key and security rules as the source GuardPoint.

If you backup encrypted data from files inside GuardPoints with LDT policies, LDT imposes a hard requirement on the backup application, or process, to backup or restore LDT metadata. The LDT metadata is stored as an extended attribute on Linux and as alternate data streams on files on Windows platforms. If the backup process cannot backup the metadata, then LDT protected GuardPoints must backed up in clear key.

Customers are required to verify their backup application, or process, to ensure extended attributes (on Linux) or alternate data streams (on Windows) are backed up and restored through their backup/restore processes.

Note: Make sure that you suspend LDT operations before you start the backup process, and resume them after the backup process completes.

Learn Mode

Learn Mode provides a temporary method for disabling the blocking behavior of VTE/LDT policies. While useful for quality assurance, troubleshooting, and mitigating deployment risk, Learn Mode is not intended to be enabled permanently for a policy in production. This prevents the policy Deny rules from functioning as designed in the policy rule set.

Ensure that the policy is properly configured for use in Learn Mode. Any Security Rule that contains a Deny effect must have Apply Key applied as well. This is to prevent data from being written in mixed states, resulting in the loss of access or data corruption.

Note: Apply Key will have no effect when combined with a Deny rule unless the policy is in Learn Mode.

QoS

By default, the QoS component of LDT does not monitor live transformation operations unless you enable QoS on your host by entering QoS parameters on the DSM. To avoid overhead of LDT on your production system, you must select QoS parameters suitable to your production environment. Refer to the QoS section of LDT guide for information on tuning QoS. It is critical that you understand the impact of live data transformation on your system and how to manage this impact using QoS.

Upgrade to VTE v5.3.0

Since the the first release of the Live Data Transformation feature, Thales e-Security has made several improvements in the areas of error handling, interoperability with applications, and Quality of Service (QoS). Thales e-Security strongly recommends that new deployments use version 6.3.0 (or later) of VTE. Thales e-Security further recommends that customers who already deployed the Live Data Transformation feature to upgrade to version 6.3.0.

Windows Platform

File Handling

The LDT process is subjected to all of the File System policies and attributes set on the files. In some cases, this prevents LDT from encrypting a file. If users or applications are accessing files while LDT is in progress, LDT cannot change the attributes of the files and encrypt the file. It is critical that you understand how LDT handles various types of files:

- **NTFS Encryption and Compression**

  If NTFS encryption or compression is enabled on a file or folder, the LDT process cannot encrypt these files. To maintain the data coherency, LDT skips the encryption of the these files. These files display as “passthrough” files in the LDT statistics.
- **Read-Only Files**

  As the LDT process performs a read-encrypt-write operation on a file, it cannot encrypt read-only files. The LDT process skips these files and changes to the INCOMPLETE state.
- **Executable Files**

  If a executable is running or files are exclusively locked by the application, the LDT process cannot encrypt those files as it is unable to acquire the required locks on the files. LDT skips these files and changes to the INCOMPLETE state.

File Modification

The LDT process performs a read-encrypt-write operation on the files that need to be encrypted, (also known as rekeying). Previously, file modification and access dates were changed when LDT was processing. In order to maintain compatibility for applications, we addressed this issue by saving a copy of the original access and modification times, and restoring them after the encryption completed. Preserved timestamps are updated during the rekey process, if an application/user accesses the files during rekey.

Note: Thales strongly recommends that you upgrade to the newly released v6.1.0 so that the access time and modification time is restored correctly.

LDT Limitations

- LDT on CIFS is not supported.
- LDT on a ReFS file system runs slowly because of limited support from the Extended Attributes on the ReFS file system.

Note: Customers running older versions of ReFS.sys on Windows Server 2012 R2 should be aware of the memory growth issue encountered by the Thales engineering team. This issue seems to occur only when LDT is running on a large number of files. As the system memory consumption by REFS file system increases, it can eventually make the system unresponsive. This issue does not occur with the recent versions of ReFS file system drivers available on Windows Server 2016. After consulting with Microsoft, they suggest that all customers migrate to Windows Server 2016 if they are using ReFS file.

Logical Sector Size

LDT Windows transformation is supported if the **logical sector size** is more than 512 Bytes. (A logical sector size of 4K is supported.) To find the logical sector size of the file system, type:

```
> fsutil fsinfo ntfsInfo <volume pathname>
```

Example:

```
> fsutil fsinfo ntfsInfo C:
```

System Response:

```
NTFS Volume Serial Number :        0x5092568a92567506
NTFS Version   :                   3.1
LFS Version    :                   2.0
Number Sectors :                   0x000000001c004eeb
Total Clusters :                   0x00000000038009dd
Free Clusters  :                   0x00000000008bb274
Total Reserved :                   0x0000000000001864
Bytes Per Sector  :                512
.
.
.
```

Upgrade Notes

CAUTION
Do not upgrade the Windows agent without valid DSM connectivity.

During the upgrade, InstallShield cleans up the agent configuration on the host. If the agent does not have DSM connectivity, then it cannot pull the configuration from the DSM after it reboots. As the configuration is cleaned up, the Windows agent removes all of the GuardPoints from the host.

VSS Volumes

If a Shadow Copy volume is present on a volume before LDT starts, and this volume is accessed by the application or user, then Thales VTE reads the LDT metadata from the shadow copy volume snapshot and incorrectly uses this metadata for the original file. Because the metadata on the snapshot is old, this may lead to double encryption of the files inside the GuardPoint.

Thales highly recommends that, if you are running a prior version (earlier than 6.0.3.68) of VTE, then to avoid this issue, you must delete all of the old VSS volumes where LDT GuardPoints exist.

Linux Platform

LDT Limitations (Linux)

- LDT on NFS share is not supported.
- LDT support is limited to ext3, ext4, XFS, and VxFS file systems when `user_xattr` mount option is enabled.

Americas
Arboretum Plaza II, 9442 Capital of Texas Highway North,
Suite 100, Austin, TX 78759 USA
Tel: +1 888 343 5773 or +1 512 257 3900
Fax: +1 954 888 6211 | E-mail: sales@thalesesec.com

Asia Pacific – Thales Transport & Security (HK) Ltd
Unit 4101-3, 41/F, Sunlight Tower, 248 Queen's Road East
Wanchai, Hong Kong | Tel: +852 2815 8633
Fax: +852 2815 8141 | E-mail: asia.sales@thales-esecurity.com

Europe, Middle East, Africa
Meadow View House, Long Crendon,
Aylesbury, Buckinghamshire HP18 9EQ
Tel: +44 (0)1844 201800 | Fax: +44 (0)1844 208550
E-mail: emea.sales@thales-esecurity.com

> thalescpl.com <