Quantum

User's Guide User's Guide User's Guide User's Guide User's Guide

Quantum Encryption Key Manager

Scalar i-Series

6-01847-01

Quantum Encryption Key Manager User's Guide, 6-01847-01, Rev A01, November 2007.
Product of USA.

COPYRIGHT STATEMENT

TRADEMARK STATEMENT

Contents

Figures

Preface

Audience

This book is intended for storage and security administrators responsible for security and backup of vital data, and anyone assisting in the setup and maintenance of Quantum Encryption Key Manager (Q-EKM) servers in the operating environment. It assumes the reader has a working knowledge of storage devices and networks.

Purpose

This book contains information to help you use the Q-EKM component for the Java™ platform. It includes concepts and procedures pertaining to:

- Encryption on the IBM LTO Ultrium 4 tape drives
- Cryptographic keys
- Digital certificates

Document Organization

This document is organized as follows:

- Chapter 1, Tape Encryption Overview, provides an overview of tape encryption and the Quantum Encryption Key Manager (Q-EKM) components.
- Chapter 2, Planning Your Q-EKM Environment, provides the information you need and the factors you should consider when determining the best configuration for your Q-EKM environment.
- Chapter 3, Passwords and Commands, provides the operational procedures for using Q-EKM.
- Chapter 4, Troubleshooting, provides troubleshooting procedures for common Q-EKM issues.
- Appendix A, Setting the System Path Variable in Windows, tells you how to set the system path so you can enter Q-EKM commands from the command line without changing the directory to the Q-EKM directory.

This document also has a glossary and an index.

Notational Conventions

This manual uses the following conventions:

Note: Notes emphasize important information related to the main topic.

Caution: Cautions indicate potential hazards to equipment and are included to prevent damage to equipment.

Warning: Warnings indicate potential hazards to personal safety and are included to prevent injury.

This manual also uses the following conventions:

Convention	Usage
bold	**Bold** words or characters represent system elements that you must use literally, such as command names, file names, flag names, path names, and selected menu options.
Arial regular text	Examples, text specified by the user, and information that the system displays appear in Arial regular font.
italic	*Italicized* words or characters represent variable values that you must supply.
[item]	Indicates optional items.
{item}	Encloses a list from which you must choose an item in format and syntax descriptions.
|	A vertical bar separates items in a list of choices.
<key>	Indicates keys you press.

Related Documents

The following publications provide information related to encryption on Scalar® libraries:

Document No.	Document Title
6-01210-xx	Scalar i500 User's Guide
6-01601-xx	Setting Up Encryption Key Management On Your Scalar i500 Library
6-00421-xx	Scalar i2000 User's Guide
6-01244-xx	Scalar i2000 User's Guide Addendum

Refer to the appropriate product manuals for information about your tape drive and cartridges.

Contacts

Quantum company contacts are listed below.

Quantum Corporate Headquarters

To order documentation on Quantum Encryption Key Manager or other products contact:

Quantum Corporation
P.O. Box 57100
Irvine, CA 92619-7100
(949) 856-7800
(800) 284-5101

Technical Publications

To comment on existing documentation send an e-mail to:

doc-comments@quantum.com

Quantum Home Page

Visit the Quantum home page at:

http://www.quantum.com

Getting More Information or Help

StorageCare™, Quantum's comprehensive service approach, leverages advanced data access and diagnostics technologies with cross-environment, multi-vendor expertise to resolve backup issues faster and at lower cost.

Accelerate service issue resolution with these exclusive Quantum StorageCare services:

- **Service and Support Web site** - Register products, license software, browse Quantum Learning courses, check backup software and operating system support, and locate manuals, FAQs, firmware downloads, product updates and more in one convenient location. Benefit today at: http://www.quantum.com/ServiceandSupport/Index.aspx.

- **eSupport** - Submit online service requests, update contact information, add attachments, and receive status updates via email. Online Service accounts are free from Quantum. That account can also be used to access Quantum's Knowledge, a comprehensive repository of product support information. Sign up today at: http://www.quantum.com/ServiceandSupport/eSupport/Index.aspx.

For further assistance, or if training is desired, contact a Technical Assistance Center:

North America and Mexico	+1 800-284-5101
Europe, Middle East, and Africa	00800-9999-3822
Worldwide support:	http://www.quantum.com/ServiceandSupport/Contacts/Worldwide/Index.aspx

For the most up to date information on Quantum Global Services, please visit: http://www.quantum.com/ServiceandSupport/Contacts/Worldwide/Index.aspx.

Non-Quantum Support

Red Hat Information

The following URL provides access to information about Red Hat Linux® systems:

- http://www.redhat.com

Microsoft Windows Information

The following URL provides access to information about Microsoft® Windows® systems:

- http://www.microsoft.com

Chapter 1

Tape Encryption Overview

Data is one of the most highly valued resources in a competitive business environment. Protecting that data, controlling access to it, and verifying its authenticity while maintaining its availability are priorities in our security-conscious world. Data encryption is a tool that answers many of these needs.

The IBM LTO-4 Fibre Channel or SAS tape drive is capable of encrypting data as it is written to any LTO-4 data cartridge. Encryption is performed at full line speed in the tape drive after compression. (Compression is more efficiently done before encryption.) This new capability adds a strong measure of security to stored data without the processing overhead and performance degradation associated with encryption performed on the server or the expense of a dedicated appliance.

Tape Drive Encryption Solution

Four major elements comprise the tape drive encryption solution:

- Encryption-Enabled Tape Drive
- Encryption Key Management (EKM)
- Encryption Policy
- Encryption-Enabled Tape Library

Encryption-Enabled Tape Drive

IBM LTO-4 Fibre Channel and SAS tape drives are *encryption-capable*. This means that they are functionally capable of performing hardware encryption, but this capability has not yet been activated. In order to perform hardware encryption, the tape drives must be *encryption-enabled*. They can be encryption enabled via the tape library.

SCSI IBM LTO-4 tape drives are *encryption aware* (they can load and handle encrypted LTO-4 cartridges, but cannot process encryption operations).

Encryption Key Management (EKM)

Encryption involves the use of several kinds of keys, in successive layers. How these keys are generated, maintained, controlled, and transmitted depends upon the operating environment where the encrypting tape drive is installed. Some applications, such as Tivoli Storage Manager (TSM), are capable of performing key management. For environments without such applications or those where application agnostic encryption is desired, Quantum provides the Quantum Encryption Key Manager (Q-EKM) component for the Java platform to perform all necessary key management tasks. Managing Encryption With Q-EKM on page 4 describes these tasks in more detail.

Encryption Policy

The encryption policy is the method used to implement encryption. It includes the rules that govern which volumes are encrypted and the mechanism for key selection. See Managing Encryption With Q-EKM on page 4 for more information.

Encryption-Enabled Tape Library

On an encryption-enabled library, tape encryption occurs automatically and transparently. The library communicates with the EKM server to obtain encryption keys for the drives to read from or write to encrypted data to the tapes.

Quantum Encryption Key Manager (Q-EKM) Components

Q-EKM is part of the IBM Java environment and uses the IBM Java Security components for its cryptographic capabilities. Q-EKM has three main components that are used to control its behavior:

- Keystore
- Configuration Files
- Tape Drive Table

Keystore

The keystore is defined as part of the Java Cryptography Extension (JCE) and an element of the Java Security components, which are, in turn, part of the Java runtime environment. Q-EKM supports the JCEKS keystore.

The keystore holds the certificates and keys used by Q-EKM to perform cryptographic operations.

The keystore file is named EKMKeys.jck and is located in the root QEKM folder as follows:

- **Windows:** C:\Program Files\Quantum\QEKM
- **Linux:** opt/Quantum/QEKM

Caution: It is impossible to overstate the importance of preserving your keystore data. Without access to your keystore, you will not be able to decrypt your encrypted tapes. Please see Backing Up Keystore Data on page 12 and Disaster Recovery Planning on page 12 for information on how to protect your keystore data.

Configuration Files

The configuration files contain the setup for the Q-EKM installation. The two configuration files are named:

- ClientKeyManagerConfig.properties
- KeyManagerConfig.properties

The configuration files are located in the root QEKM folder as follows:

- **Windows:** C:\Program Files\Quantum\QEKM
- **Linux:** opt/Quantum/QEKM

Caution: Do not edit these files. If you make a mistake when altering the configuration files, you could lose access to your keystore and be unable to encrypt or restore data.

Tape Drive Table

The tape drive table is used by Q-EKM to keep track of the tape devices it supports. The tape drive table is a noneditable, binary file whose location is specified in the configuration file. Q-EKM automatically adds new/replaced tape drives to the drive table.

Managing Encryption With Q-EKM

The Quantum Encryption Key Manager (Q-EKM) component for the Java platform is a Java software program that assists IBM encryption-enabled tape drives in generating, protecting, storing, and maintaining encryption keys that are used to encrypt information being written to, and decrypt information being read from, tape media (tape and cartridge formats).

Q-EKM operates on Windows Server 2003 and Red Hat Enterprise Linux 4, and is designed to be a shared resource within an Enterprise.

Q-EKM uses a keystore to hold JCEKS keys and certificates required for all encryption tasks.

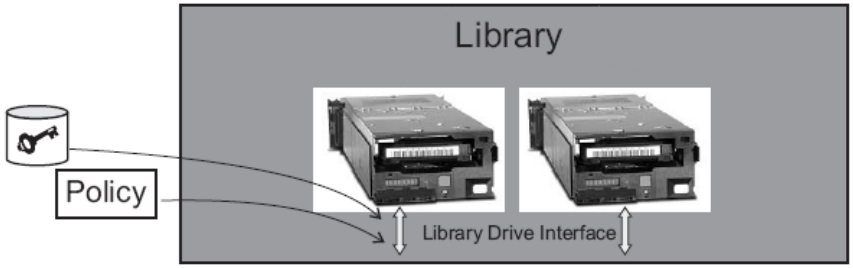
Q-EKM acts as a process awaiting key generation or key retrieval requests sent to it through a TCP/IP communication path between Q-EKM and the tape library.

When a tape drive writes encrypted data, it first requests an encryption key from Q-EKM.

Upon receipt of the request, Q-EKM retrieves an existing Advanced Encryption Standard (AES) key from a keystore and wraps it for secure transfer to the tape drive, where it is unwrapped upon arrival and used to encrypt the data being written to tape.

When an encrypted tape is read by an IBM LTO-4 tape drive, Q-EKM retrieves the required key from the keystore, based on the information in the Key ID on the tape, and serves it to the tape drive wrapped for secure transfer.

Figure 1 Encryption Policy Engine and Key Manager Locations



The Library Layer is the enclosure for tape storage, such as the Quantum Scalar i500 or Scalar i2000 tape library, and contains an internal interface to each tape drive within it.

Library-Managed Encryption

Library-Managed tape encryption is provided for IBM LTO-4 tape drives in a Quantum Scalar i500 or Scalar i2000 tape library. Key generation and management is performed by Q-EKM. Policy control and keys pass through the library-to-drive interface, making encryption transparent to applications.

Encryption Keys

An encryption key is typically a random string of bits generated specifically to scramble and unscramble data. Encryption keys are created using algorithms designed to ensure that each key is unique and unpredictable. The longer the length of key used, the harder it is to break the encryption code.

The IBM LTO-4 method of encryption uses 256-bit AES algorithm keys to encrypt data. 256-bit AES is the encryption standard currently recognized

and recommended by the US government, which allows three different key lengths. 256-bit keys are the longest allowed by AES.

Two types of encryption algorithms may be used by Q-EKM:

- Symmetric algorithms
- Asymmetric algorithms

Symmetric, or secret key encryption, uses a single key for both encryption and decryption. Symmetric key encryption is generally used for encrypting large amounts of data in an efficient manner. 256-bit AES keys are symmetric keys.

Asymmetric, or public/private encryption, uses a pair of keys. Data that is encrypted using one key can only be decrypted using the other key in the public/private key pair. When an asymmetric key pair is generated, the public key is typically used to encrypt, and the private key is typically used to decrypt.

Q-EKM uses both symmetric and asymmetric keys—symmetric encryption for high-speed encryption of user or host data, and asymmetric encryption (which is necessarily slower) for protecting the symmetric key.

Upon installation, Q-EKM generates 1024 unique encryption keys.

Encryption Key Processing

In library-managed tape encryption, unencrypted data is sent to the IBM LTO-4 tape drive and converted to ciphertext using a pre-generated symmetric data key from the keystore available to Q-EKM, and is then written to tape.

Q-EKM selects a pre-generated data key in round-robin fashion. Data keys are reused on multiple tape cartridges when all pre-generated data keys have been used at least once.

The data key is sent to the IBM LTO-4 tape drive in encrypted, or *wrapped*, form by Q-EKM. The IBM LTO-4 tape drive unwraps this data key and uses it to perform encryption or decryption. However, no wrapped key is stored anywhere on the IBM LTO-4 tape cartridge.

After the encrypted volume is written, the data key must be accessible, based on the alias or key label, and available to Q-EKM in order for the volume to be read.

Chapter 2

Planning Your Q-EKM Environment

Use the information in this chapter to determine the best Q-EKM configuration for your needs. Many factors must be considered when you are planning how to set up your encryption strategy. Please review these topics with care.

System Requirements

Server Requirements

Q-EKM server requirements are:

- Xeon-class server.
- Minimum 1 GB memory.
- Minimum 10 GB free hard disk space.
- The Q-EKM server must have IP connectivity through any firewalls to all Quantum libraries using the Q-EKM server to obtain LTO-4 encryption keys. The Q-EKM firmware uses TCP port 3801 for the Q-EKM server and TCP port 443 for SSL, by default.
- The Q-EKM server should be protected and backed up following your data protection practices so that critical keystore data can be quickly restored in the event of a server failure.

Operating System Requirements

Q-EKM runs on either:

- Windows Server 2003
- Red Hat Enterprise Linux 4

Supported Tape Drives

Q-EKM supports the following tape drives:

Scalar i500 tape library	IBM LTO-4 (Fibre-Channel and SAS)
Scalar i2000 tape library	IBM LTO-4 (Fibre-Channel only)

Library Firmware Requirements

Following are the minimum library firmware requirements needed to run Q-EKM:

Scalar i500 tape library	410G.GS007 firmware code
Scalar i2000 tape library	Please download the latest code available for the i6.1 release or later.

Tape Drive Firmware Requirements

Following are the minimum tape drive firmware requirements needed to run Q-EKM:

IBM LTO-4 tape drives on the Scalar i500 tape library	77BA drive code for SP4 77BE drive code for SP4.2 and beyond
IBM LTO-4 tape drives on the Scalar i2000 tape library	77BE drive code

Using Multiple Q-EKM Servers for Redundancy

Q-EKM is designed to work with tape drives and libraries to allow redundancy, and thus high availability, so you can have more than one Q-EKM server servicing the same tape drives and libraries. Moreover, these Q-EKM servers need not be on the same systems as the tape drives and libraries. The only requirement is that they be available to the libraries through TCP/IP connectivity.

This allows you to have two Q-EKM servers that are mirror images of each other with built-in synchronization and back up of the critical keystore information as well as a failover in the event that one Q-EKM server becomes unavailable. When you configure your library, you can point it to two Q-EKM servers (primary and secondary). If the primary Q-EKM server becomes unavailable for any reason, the library will use the secondary Q-EKM server.

At this time, Q-EKM supports up to two servers accessing a single keystore (rather than a separate keystore for each server). In order for the secondary server to be a backup, the keystore must be identical to that of the primary server.

Q-EKM Server Configurations

Q-EKM can be installed on a single server or on two servers.

Single-Server Configuration

A single-server configuration, shown in figure 2, is the simplest Q-EKM configuration. However, because of the lack of redundancy, it is not recommended. In this configuration, all tape drives rely on a single key manager server with no backup. Should the server go down, the keystore becomes unavailable, making any encrypted tape unreadable (and preventing encrypted writes). In a single-server configuration, you must make sure that current, non-encrypted backup copies of the keystore and configuration files are maintained in a safe place, separate from Q-EKM, so its function can be rebuilt on a replacement server if the server copies are lost.

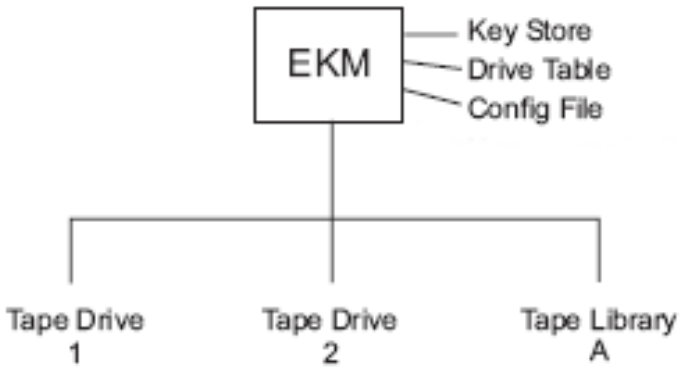
The keystore and configuration files are:

- ClientKeyManagerConfig.properties
- KeyManagerConfig.properties
- EKMKeys.jck

The files are all in the root QEKM folder as follows:

- **Windows:** C:\Program Files\Quantum\QEKM
- **Linux:** opt/Quantum/QEKM

Figure 2 Single Q-EKM Server



Two-Server Configuration

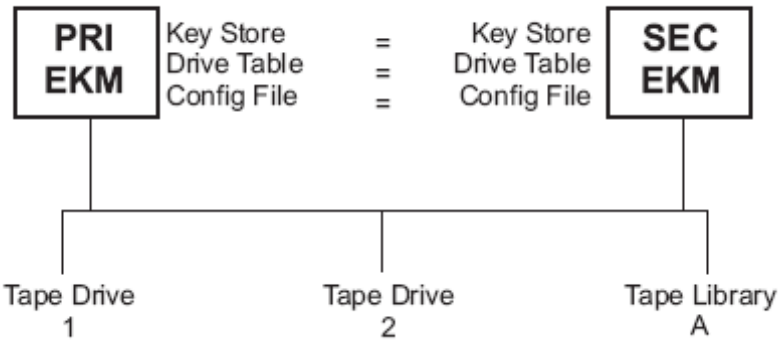
The recommended two-server configuration allows the library to automatically failover to the secondary Q-EKM server should the primary Q-EKM server be inaccessible for any reason.

> **Note:** When different Q-EKM servers are used to handle requests from the same set of tape drives, the information in the associated keystores MUST be identical. This is required so that regardless of which Q-EKM server is contacted, the necessary information is available for the Q-EKM server to support requests from the tape drives.

In an environment with two Q-EKM servers, such as those shown in Figure 3, the library will automatically failover to the secondary Q-EKM server should the primary go down. In such a configuration it is essential that the two Q-EKM servers share the same keystore file and that the servers are synchronized (Quantum Global Services can set up synchronization when they install your server).

Once synchronization is configured, updates to the configuration files and drive table of the primary Q-EKM server are automatically duplicated on the secondary Q-EKM server. However, the keystore file is not automatically updated. After any change to the keystore on the primary server (adding, importing, or exporting keys) the keystore file on the primary server must be manually copied to the secondary server.

Figure 3 Two Q-EKM Servers With Shared Configurations



Backing Up Keystore Data

Due to the critical nature of the keys in the keystore, you should always back up this data so that you can recover it, as needed, and be able to read the tapes that were encrypted using those certificates associated with that tape drive or library.

Use your system backup capabilities to back up the entire QEKM directory regularly. The QEKM directory is located here:

- **Windows:** C:\Program Files\Quantum\QEKM
- **Linux:** opt/Quantum/QEKM

Caution: Do not use Q-EKM to encrypt the backups! Back up to clear tape! If you encrypt your backup, and you later lose your keystore, you will not be able to decrypt the tapes to recover your data.

In addition, it is recommended that you maintain a primary and secondary Q-EKM server and keystore copy (for backup as well as failover redundancy). See Single-Server Configuration on page 9 for more information.

For disaster recovery, see Disaster Recovery Planning on page 12.

Disaster Recovery Planning

Quantum recommends that you plan for disaster recovery in the event that your primary and secondary servers become unavailable.

Disaster recovery requires that the you maintain, in a secure location, current, non-Q-EKM encrypted copies of the following three files:

- ClientKeyManagerConfig.properties
- KeyManagerConfig.properties
- EKMKeys.jck

The files are all in the root QEKM folder as follows:

- **Windows:** C:\Program Files\Quantum\QEKM
- **Linux:** opt/Quantum/QEKM

Successful recovery requires the following two things:

- The copied files must be current. Any time the keystore or configuration files are changed (i.e., creating, importing, or exporting keys or certificates), you must remember to save a backup. If you back up your files regularly as recommended, this should not be an issue (see Backing Up Keystore Data on page 12).
- The backup files must not be encrypted with Q-EKM. If the primary and secondary servers are unavailable, the encrypted files will not be able to be decrypted and reused in the disaster recovery server.

Upon failure of the Q-EKM server, Quantum Global Services can set up a new "disaster recovery" Q-EKM server or servers to replace the ones that became unavailable. Setup of the new server includes copying the three files listed above onto the new server.

Considerations for Sharing Encrypted LTO-4 Tapes Offsite

> **Note:** Sharing keys between keystores is not supported at initial launch.

It is common practice to share tapes with other organizations (that are not using the same Q-EKM server/keystore for encryption) for data transfer, joint development, contracting services, or other purposes.

Q-EKM creates unique key aliases across all Q-EKM installations worldwide. This ensures that you can safely share Q-EKM-encrypted tapes with other sites or companies.

Note: It is important to verify the validity of any certificate received from a business partner by checking the chain of trust of such a certificate back to the Certificate Authority that ultimately signed it. If you trust the Certificate Authority, then you can trust that certificate. Alternately, the validity of a certificate can be verified if it was securely guarded in transit. Failure to verify a certificate's validity in one of these ways may open the door to a "Man-in-the-Middle" attack.

In order to share encrypted data on an IBM LTO-4 tape, a copy of the symmetric key used to encrypt the data on the tape must be made available to the other organization to enable them to read the tape.

In order for the symmetric key to be shared, the other organization must share their public key with you. This public key will be used to wrap the symmetric key when it is exported from the Q-EKM keystore (see Export Keys on page 19).

When the other organization imports the symmetric key into their Q-EKM keystore, it will be unwrapped using their corresponding private key (see Import Keys on page 19). This ensures that the symmetric key will be safe in transit since only the holder of the private key will be able to unwrap the symmetric key.

With the symmetric key that was used to encrypt the data in their Q-EKM keystore, the other organization will then be able to read the data on the tape.

Chapter 3

Passwords and Commands

This chapter provides information about the passwords and commands used on Q-EKM.

Passwords

There are two different passwords you need to use with Q-EKM. They are:

- Command Password
- Keystore Password

Command Password

You use the command password when issuing commands via the command line. Every time you issue a command to Q-EKM, you must enter the command password. The default command password is **changeME**. You can change this password using the **chgpwdQEKMServer** command (see Change Command Password on page 17).

The command password is case sensitive, can contain a maximum of 24 characters, and can contain any combination of letters, numbers, and special characters (for example, !@#$%^&*) .

Keystore Password

The keystore password allows you to add, import, and export keys or certificates to the keystore (EKMkeys.jck).

Quantum Global Services sets up your keystore password at the initial Q-EKM server installation. The keystore password is case sensitive, must contain a minimum of 6 characters and a maximum of 24 characters, and can contain any combination of letters, numbers, and special characters (for example, !@#$%^&*) .

Encryption and decryption of tapes will still occur if you forget the password, but if you want to read encrypted tapes provided by another organization or company, of if you want to enable other organizations or companies to read your encrypted tapes, you will need to import and export keystore information, which you cannot do without the keystore password. *If you forget the keystore password, there is no way to recover it.*

Caution: It is CRITICAL that you remember the keystore password! If you forget the password, neither you nor Quantum will be able to recover it. You will also not be able to change the keystore, including adding, importing, or exporting keys and certificates. Quantum recommends that you make note of the keystore password and store it in an accessible location, and make sure more than one person knows what the password is.

Commands

Q-EKM provides a command set that can be issued from the Q-EKM server command prompt.

If you are using Windows, you must be in the correct directory. At the command prompt, ensure you are in the root QEKM directory by changing the directory to C:\Program Files\Quantum\QEKM. (Alternatively, you may choose to update your Windows system path variable — see Appendix A, Setting the System Path Variable in Windows.)

Note: Commands are case sensitive on Linux servers. Commands are not case sensitive on Windows servers.

Caution: These commands shut down and then restart the Q-EKM server process. Do not perform these commands if backup operations are in process.

Change Command Password

chgpwdQEKMServer

The Change Password command changes the Command Password (not to be confused with the Keystore Password). The default command password is **changeME**.

1. At the command prompt, type **chgpwdQEKMServer** and press <Enter>.
2. When prompted, enter a new command password and press <Enter>.
3. When prompted, enter the old password and press <Enter>.

   You receive confirmation that the password was successfully changed.

Change Port Settings on Q-EKM Server

portChgQEKMServer

The Q-EKM server(s) are set up with the following default TCP ports:

- **TCP Port (also referred to as the EKM Port) – Default Value 3801**. This port enables communication between the Q-EKM server and the library.
- **SSL Port – Default Value 443**. This port enables communication between Q-EKM servers (used for synchronization).

If you want to change either of the port numbers, do the following:

1. At the command prompt, enter **portChgQEKMServer**.

   The Q-EKM server stops and you are asked for the EKM user password.
2. Enter the command password.
3. When asked if the Q-EKM server was stopped, enter **y**.

4 When prompted, enter **ssl** (to change the SSL port) or **tcp** (to change the EKM port) .

5 When prompted, enter the new port number.

6 Remember that if you change the TCP (EKM) port number on the Q-EKM server, you must also change the reference to it on the library interface. See the following documents for instructions on how to do this:

- **Scalar i500:** Refer to *Setting Up Encryption Key Management On Your Scalar i500 Library* (p/n 6-01601-xx).
- **Scalar i2000:** Refer to the *Scalar i2000 User's Guide Addendum* (p/n 6-01244-04). Upon the i7 release, refer to the *Scalar i2000 User's Guide* (p/n 6-00421-xx).

Note: In order to synchronize properly, the TCP (EKM) and SSL ports on the primary and secondary Q-EKM servers must be set to the same values. Synchronization causes the entire configuration properties files of the primary server to overwrite the configuration files on the secondary server. Because the TCP (EKM) and SSL ports are listed in the configuration properties files, the primary and secondary servers must use the same TCP (EKM) and SSL port settings. Make sure the libraries that access these servers have their TCP (EKM) port configuration settings set correctly.

Debug Off and On

debugOnQEKMServer

debugOffQEKMServer

The Debug On (**debugOnQEKMServer**) command allows the debug log to capture all Q-EKM communication.

The Debug Off (**debugOffQEKMServer**) command prevents the debug log from capturing data.

The debug log (path and filename: QEKM\keymanager**debug_server**) captures TCP/SSL communication between the EKM server and the drives (crypto class, raw sense data, port number/server IP, get drive ID/ checking IP and vendor, etc.). This file will continue to grow as long as debug is turned on. It can grow very quickly.

Debug is turned off by default in order to prevent the debug log file from becoming too large and overwhelming the system. If you encounter a problem that requires assistance from Quantum Global Services, you will probably need to turn debug on and then re-create the problem in order to generate troubleshooting data.

> **Note:** **Remember to turn debug off once you have finished gathering data.** (If you forget to do this and the file becomes too large, stop the Q-EKM server, delete the **debug_server** file, and restart the Q-EKM server. This re-creates the debug log with no data in it. You can then turn debug on or off as needed.)

The Debug On and Debug Off commands shut down the Q-EKM server, change the KeyManagerConfig.properties file, and then restart the Q-EKM server.

Export Keys

Not available at this time.

Import Keys

Not available at this time.

List Drives

listDrivesQEKMServer

This command returns a list of all drives that have ever successfully asked for a key from the Q-EKM server. Quantum's standard Q-EKM installation automatically adds drives to the drive table. There is no maximum number of entries, and all drives will remain on the list even if they are removed from the library.

1. At the prompt, type **listDrivesQEKMServer** and press <Enter>.
2. When prompted for the password, enter the default command password **changeME** (or, if you have changed the password, enter the new password).

   The returned information listed looks similar to the following:

   ```
   Drive entries: 2

   SerialNumber = 001300000392

   SerialNumber = 001310000363
   ```

Start Q-EKM Server

startQEKMServer

The Start Q-EKM Server command starts the Q-EKM server.

1. At the prompt, type **startQEKMServer** and press <Enter>.

   You receive the following message:

   ```
   Starting EKM Server...
   Please check the logs to make sure EKM Server has started
   successfully.
   ```

2. To verify the Q-EKM server started, you can check the native_stderr.log and native_stdout.log files (located in the keymanager folder in the QEKM directory), or you can use the **statusQEKMServer** command (see Status).

Status

statusQEKMServer

The Status command displays whether Q-EKM server is started or stopped.

1. At the prompt, type **statusQEKMServer** and press <Enter>. It may take a minute for the next prompt to appear.
2. When prompted for the password, enter the default command password **changeME** (or, if you have changed the password, enter the new password)
   - If the Q-EKM server is running, you receive confirmation that looks similar to the following:

     ```
     Server is running. TCP port: 3801, SSL port: 443
     ```

   - If the Q-EKM server is not running, you receive the following:

     ```
     EKM server cannot be reached. It appears to be stopped.
     ```

Stop Q-EKM Server

stopQEKMServer

The Stop Q-EKM server command stops the Q-EKM server.

1. At the prompt, type **stopQEKMServer** and press <Enter>.
2. When prompted for the password, enter the default command password **changeME** (or, if you have changed the password, enter the new password).

You receive confirmation that looks similar to the following:

```
EKMServer:  shut down complete.
```

Synchronize Servers

syncQEKMServer

The Synchronize Servers command enables synchronization of the primary and secondary Q-EKM servers. Synchronization copies the configuration files from the primary server to the secondary server automatically every hour as long as both servers are up and running and connected to the network.

Generally, Quantum Global Services will set this up for you upon Q-EKM installation and you should not need to use this command.

> **Note:** This command does not perform an "instant" or "manual" synchronization. It takes one hour for the first synchronization to occur; then, automatic synchronizations occur at one-hour intervals.

Version

versionQEKMServer

The Version command provides the version of the Q-EKM server that is currently installed.

1. At the command prompt, type **versionQEKMServer** and press <Enter>.
2. When prompted for the password, enter the default command password **changeME** (or, if you have changed the password, enter the new password).

   You receive the version information that looks similar to the following:

   ```
   Quantum EKM Server Version: 2.1_007
   ```

Chapter 4

Troubleshooting

Log Files

Q-EKM provides the following log files that can be used for troubleshooting and debug:

- Audit Log
- Debug Log
- Standard Error Messages Log
- Standard Out Messages Log

Audit Log

The audit log timestamps drive vendor, serial number, world-wide name (WWN), media volser, and key servings results. The data it collects is a subset of the much more comprehensive Debug Log. The audit log is always available and collecting data. The Q-EKM application limits the size of this file to 10 MB. When the file reaches the maximum size, old information is deleted as new information is added.

The audit log path and file name are:

- **Windows:**
  C:\Program Files\Quantum\QEKM\keymanager\audit**kms_audit.log**
- **Linux:** opt/Quantum/QEKM/keymanager/audit/**kms_audit.log**

Debug Log

The debug log captures TCP/SSL communication between the Q-EKM server and the drives (crypto class, raw sense data, port number/server IP, get drive ID/checking IP and vendor, etc.). The debug log does not collect information unless debug is turned on (see Debug On/Debug Off on page 36 for more information on turning debug on and off). The debug log will continue to grow as long as debug is turned on. It can grow very quickly.

Debug is turned off by default in order to prevent the debug log file from becoming too large and overwhelming the system. If you have a problem that requires assistance from Quantum Global Services, you will probably need to turn debug on and then re-create the problem in order to generate troubleshooting data.

> **Note:** **Remember to turn debug off once you have finished gathering data.** (If you forget to do this and the file becomes too large, stop the Q-EKM server, delete the **debug_server** file, and restart the Q-EKM server. This re-creates the debug log with no data in it. You can then turn debug on or off as needed.)

The debug log path and file name are:

- **Windows:** C:\Program Files\Quantum\QEKM\keymanager**debug_server**
- **Linux:** opt/Quantum/QEKM/keymanager/**debug_server**

Standard Error Messages Log

The standard error messages log lists errors that occurred during Q-EKM startup or shutdown. This log is generally used in combination with the Standard Out Messages Log.

> **Note:** There is currently a benign error message that appears in this log. It will be fixed in future versions and can be ignored. The message is:
> [Fatal Error] :-1:-1: Premature end of file.

The standard error messages log path and file name are:

- **Windows:**
  C:\Program Files\Quantum\QEKM\keymanager**native_stderr.log**
- **Linux:** opt/Quantum/QEKM/keymanager/**native_stderr.log**

Standard Out Messages Log

The standard out messages log provides information about Q-EKM startup and shutdown operations, and lets you know whether the operation completed successfully. This log is generally used in combination with the Standard Error Messages Log.

The standard out messages log path and file name are:

- **Windows:**
  C:\Program Files\Quantum\QEKM\keymanager**native_stdout.log**
- **Linux:** opt/Quantum/QEKM/keymanager/**native_stdout.log**

Errors Reported By Q-EKM

This section defines error messages that are reported by Q-EKM in the audit log (see Audit Log on page 22).

The table below includes the error number, a short description of the failure, and corrective actions.

Error Number	Description	Action
EE02	Encryption Read Message Failure: DriverErrorNotifyParameterError: "Bad ASC & ASCQ received. ASC & ASCQ does not match with either of Key Creation/Key Translation/Key Acquisition operation."	The tape drive asked for an unsupported action. Ensure that you are running the latest version of Q-EKM (to determine the latest version, contact your Quantum Representative). Check the versions of drive or library firmware and update them to the latest release, if needed. Turn on Debug on the Q-EKM server. Try to re-create the problem and gather debug logs. When finished collecting data, turn Debug off. If the problem persists, contact Quantum Global Call Center.

Error Number	Description	Action
EE0F	Encryption logic error: Internal error: "Unexpected error. Internal programming error in EKM."	Ensure that you are running the latest version of Q-EKM (to determine the latest version, contact your Quantum Representative). Check the versions of drive or library firmware and update them to the latest release, if needed. Turn on Debug on the Q-EKM server. Try to re-create the problem and gather debug logs. When finished collecting data, turn Debug off. If the problem persists, contact Quantum Global Call Center.
EE23	Encryption Read Message Failure: Internal error: "Unexpected error....."	The message received from the drive or library could not be parsed because of general error. Ensure that you are running the latest version of Q-EKM (to determine the latest version, contact your Quantum Representative). Turn on Debug on the Q-EKM server. Try to re-create the problem and gather debug logs. When finished collecting data, turn Debug off. If the problem persists, contact Quantum Global Call Center.

Error Number	Description	Action
EE25	Encryption Configuration Problem: Errors that are related to the drive table occurred.	Ensure that the config.drivetable.file.url is correct in the KeyManagerConfig.properties file, if that parameter is supplied. Run the listdrives -drivename <drivename> command on the Q-EKM server to verify whether the drive is correctly configured (for example, the drive serial number, alias, and certificates are correct). Ensure that you are running the latest version of Q-EKM (to determine the latest version, contact your Quantum Representative). Check the versions of drive or library firmware and update them to the latest release, if needed. Turn on Debug on the Q-EKM server. Try to re-create the problem and gather debug logs. When finished collecting data, turn Debug off. If the problem persists, contact Quantum Global Call Center.
EE29	Encryption Read Message Failure: Invalid signature	The message received from the drive or library does not match the signature on it. Ensure that you are running the latest version of Q-EKM (to determine the latest version, contact your Quantum Representative). Turn on Debug on the Q-EKM server. Try to re-create the problem and gather debug logs. When finished collecting data, turn Debug off. If the problem persists, contact Quantum Global Call Center.

Error Number	Description	Action
EE2B	Encryption Read Message Failure: Internal error: "Either no signature in DSK or signature in DSK can not be verified."	Ensure that you are running the latest version of Q-EKM (to determine the latest version, contact your Quantum Representative). Check the versions of drive or library firmware and update them to the latest release, if needed. Turn on Debug on the Q-EKM server. Try to re-create the problem and gather debug logs. When finished collecting data, turn Debug off. If the problem persists, contact Quantum Global Call Center.
EE2C	Encryption Read Message Failure: QueryDSKParameterError: "Error parsing a QueryDSKMessage from a device. Unexpected dsk count or unexpected payload."	The tape drive asked Q-EKM to do an unsupported function. Ensure that you are running the latest version of Q-EKM (to determine the latest version, contact your Quantum Representative). Check the versions of drive or library firmware and update them to the latest release, if needed. Turn on Debug on the Q-EKM server. Try to re-create the problem and gather debug logs. When finished collecting data, turn Debug off. If the problem persists, contact Quantum Global Center.

Error Number	Description	Action
EE2D	Encryption Read Message Failure: Invalid Message Type	Q-EKM received a message out of sequence or received a message that it does not know how to handle. Ensure that you are running the latest version of Q-EKM (to determine the latest version, contact your Quantum Representative). Turn on Debug on the Q-EKM server. Try to re-create the problem and gather debug logs. When finished collecting data, turn Debug off. If the problem persists, contact Quantum Global Call Center.
EE2E	Encryption Read Message Failure: Internal error: Invalid signature type	The message received from the drive or library does not have a valid signature type. Ensure that you are running the latest version of Q-EKM (to determine the latest version, contact your Quantum Representative). Turn on Debug on the Q-EKM server. Try to re-create the problem and gather debug logs. When finished collecting data, turn Debug off. If the problem persists, contact Quantum Global Call Center.

Error Number	Description	Action
EE31	Encryption Configuration Problem: Errors that are related to the keystore occurred.	Check the key labels that you are trying to use or configured for the defaults. If you know that you are trying to use the defaults, then run the listdrives -drivename *drivename* command on the Q-EKM server to verify whether the drive is correctly configured (for example, the drive serial number, and associated aliases/key labels are correct). If the drive in question has no aliases/key labels associated with it, then check the values of default.drive.alias1 and default.drive.alias2. If this does not help or the alias/key label exists, then turn on Debug on the Q-EKM server, gather debug logs, and contact Quantum Global Call Center. When finished collecting data, turn Debug off.
EEE1	Encryption logic error: Internal error: "Unexpected error: EK/ EEDK flags conflict with subpage."	Ensure that you are running the latest version of Q-EKM (to determine the latest version, contact your Quantum Representative). Check the versions of drive or library firmware and update them to the latest release, if needed. Turn on Debug on the Q-EKM server. Try to re-create the problem and gather debug logs. When finished collecting data, turn Debug off. If the problem persists, contact Quantum Global Call Center.

Error Number	Description	Action
EF01	Encryption Configuration Problem: "Drive not configured."	The drive that is trying to communicate with Q-EKM is not present in the drive table. Ensure that the config.drivetable.file.url is correct in the KeyManagerConfig.properties file, if that parameter is supplied. Run the listdrives command to check whether the drive is in the list. If not, configure the drive manually by using the adddrive command with the correct drive information or set the "drive.acceptUnknownDrives" property to true using the modconfig command. Turn on Debug on the Q-EKM server. Try to re-create the problem and gather debug logs. When finished collecting data, turn Debug off. If the problem persists, contact Quantum Global Call Center.

Appendix A

Setting the System Path Variable in Windows

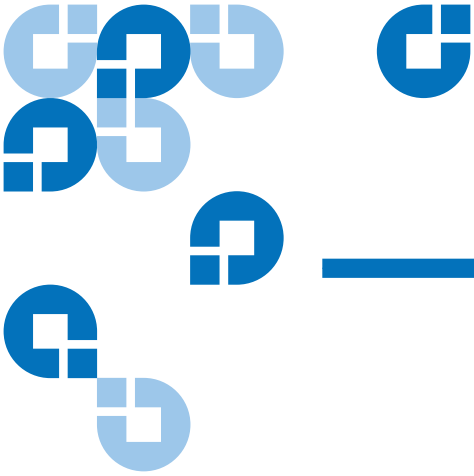
You may wish to update your system Path environment variable to include the path to the QEKM folder. This allows you to enter Q-EKM commands on any command line rather than having to change the directory to the QEKM directory each time.

To update the system Path environment variable:

1. Choose **Start > Control Panel**, then double-click **System**.
2. Select the **Advanced** tab.
3. Click **Environment Variables**.
4. Under System variables, select **Path**.
5. Click **Edit**.
6. In the Edit System Variable dialog box, click in the **Variable** value field and enter **c:\Program Files\Quantum\QEKM**.

   Note: If there is already a value in the field, use a semicolon (;) to separate the paths.

7. Click **OK, OK, OK**.



Glossary

This glossary defines the special terms, abbreviations, and acronyms used in this publication and other related publications.

A

AES Advanced Encryption Standard. A block cipher adopted as an encryption standard by the US government.

alias See key label.

C

certificate A digital document that binds a public key to the identity of the certificate owner, thereby enabling the certificate owner to be authenticated.

certificate label See key label.

D

DK Data Key. An alphanumeric string used to encrypt data.

data key An alphanumeric string used to encrypt data.

E

EKM Encryption Key Management. A system whereby encryption keys are generated, stored, protected, transferred, loaded, used, and destroyed.

encryption The conversion of data into a cipher. A key is required to encrypt and decrypt the data. Encryption provides protection

from persons or software that attempt to access the data without the key.

I

IP Internet Protocol. The method or protocol by which data is transmitted from one computer (or host) to another over the Internet using a system of addresses and gateways.

J

JCE Java Cryptography Extension.

JCEKS Java Cryptography Extension Keystore.

K

key label A unique identifier used to match the EEDK with the private key (KEK) required to unwrap the protected symmetric data key. Also called alias or certificate label depending on which keystore is used.

keystore A database of private keys and their associated X.509 digital certificate chains used to authenticate the corresponding public keys.

KS Keystore.

P

PKCS Public Key Cryptology Standards. A set of intervendor standard protocols for making possible secure information exchange on the Internet using a public key infrastructure.

private key One key in an asymmetric key pair, typically used for decryption. Q-EKM uses private keys to unwrap protected AES data keys prior to decryption.

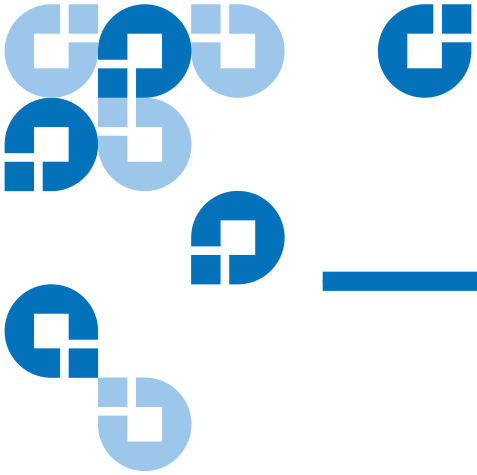
public key One key in an asymmetric key pair, typically used for encryption. Q-EKM uses public keys to wrap (protect) AES data keys prior to storing them on the tape cartridge.

Q

Q-EKM Quantum Encryption Key Manager. A Java application that handles encryption key managememt (EKM) via Quantum's tape libraries.

T

TCP Transmission Control Protocol. Works in conjunction with IP to ensure that packets reach their intended destinations.



Index

T

V