IBM Encryption Key Manager component for the Java platform

Quick Start Guide for LTO Ultrium 4

This guide gets you started with a ***basic configuration*** for encryption on LTO Gen 4 tape drives.

The IBM® Encryption Key Manager 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. These keys are used to encrypt information being written to, and decrypt information being read from, tape media (tape and cartridge formats). The Encryption Key Manager operates on z/OS®, i5/OS®, AIX®, Linux®, HP-UX, Sun Solaris, and Windows®, and is designed to be a shared resource deployed in several locations within an Enterprise. The Encryption Key Manager is capable of serving numerous IBM encrypting tape drives, regardless of where those drives reside (for example, in tape library subsystems, connected to mainframe systems through various types of channel connections, or installed in other computing systems.)

This document shows how quickly you can install the Encryption Key Manager for use with LTO Gen 4 drives and how easy it can be to set up and deploy. Because the JCEKS keystore type is the easiest and most transportable of the keystores supported, the steps below use this keystore type. If you want more information about a particular step or other supported keystore types, see the *Encryption Key Manager Introduction, Planning, and User's Guide*, which can be found at: http://www.ibm.com/support/docview.wss?&uid=ssg1S4000504.

Step 1. Download your Software

1. Be sure that a minimum of SDK 1.4.2 SR8 or SDK 5.0 SR5 is installed. If you have an earlier SDK, refer to the chapter titled "Upgrading to the latest version Encryption Key Manager or IBM Java SDK" in *Encryption Key Manager Introduction, Planning, and User's Guide* to learn how to get the latest service refresh for your software platform. Also see any upgrade notes in that chapter to learn whether a new version of Encryption Key Manager requires any configuration changes.
2. Download the IBMKeyManagementServer.jar and EKMServicesAndSamples files from this website: http://www.ibm.com/support/docview.wss?&uid=ssg1S4000504.
3. Determine whether there is a copy of the IBMKeyManagementServer.jar file in the <JAVA_INSTALL>/lib/ext directory. If so, delete it and copy in the version just downloaded.

   For i5/OS:

   a. For I/5/OS V5R3, copy the jar to /QIBM/ProdData/OS400/Java400/ext and add the following symbolic links if not already present:

   ```
   ADDLNK OBJ('/QIBM/ProdData/OS400/Java400/ext/IBMKeyManagementServer.jar')
   NEWLNK('/QIBM/UserData/Java400/ext/IBMKeyManagementServer.jar') LNKTYPE(*SYMBOLIC)

   ADDLNK OBJ('/QIBM/ProdData/OS400/Java400/ext/ibmkeycert.jar')
   NEWLNK('/QIBM/UserData/Java400/ext/ibmkeycert.jar') LNKTYPE(*SYMBOLIC)
   ```

   b. For i5/OS V5R4 and later: copy the jar to /QOpenSys/QIBM/ProdData/JavaVM/jdk50/32bit/jre/lib/ext.

Step 2. Copy the Required Files

1. Open a command window and create a directory where all of the Encryption Key Manager related files will be stored. For example:

On i5/OS

For v5R4 and later, first set set JAVA_HOME to select the J9 JVM:

```
ADDENVVAR ENVVAR(JAVA_HOME) VALUE('/QOpenSys/QIBM/ProdData/JavaVM/jdk50/32bit')
```

Open a qshell environment using either the STRQSH or QSH CL command:

```
mkdir -p /ekm/ekm1
cp /QIBM/ProdData/OS400/Java400/ext/KeyManagerConfig.properties /ekm/ekm1/KeyManagerConfig.properties
```

On Windows

```
md  c:\ekm\ekm1
```

On Unix platforms

```
mkdir   -p /var/ekm/ekm1
```

2. Copy the sample KeyManagerConfig.properties file to the same directory. For example:

   On Windows

   ```
   copy KeyManagerConfig.properties c:\ekm\ekm1\KeyManagerConfig.properties
   ```

   On Unix platforms

   ```
   cp KeyManagerConfig.properties /var/ekm/ekm1/KeyManagerConfig.properties
   ```

3. Copy the sample ClientKeyManagerConfig.properties file to the same directory. For example:

   On Windows

   ```
   copy ClientKeyManagerConfig.properties c:\ekm\ekm1\ClientKeyManagerConfig.properties
   ```

   On Unix platforms

   ```
   cp ClientKeyManagerConfig.properties /var/ekm/ekm1/ClientKeyManagerConfig.properties
   ```

4. Download the US_export_policy.jar and local_policy.jar files from https://www14.software.ibm.com/webapp/iwm/web/preLogin.do?source=jcesdk and replace the ones in your <JAVA_INSTALL>/lib/security directory. These are the unrestricted policy files the Encryption Key Manager requires in order to serve AES keys. Be sure to select ″Unrestricted JCE Policy files for SDK 1.4.2″ which works for both Java 1.4.2 and Java 5.0 SDKs. Do not select the 1.4.1 version as these are not compatible.

   Note: This step is not required on i5/OS because the unrestricted policy files are included with the operating system.

Step 3. Create a JCEKS Keystore

The Encryption Key Manager needs a keystore with a certificate and private key. This certificate will be used to secure communications among Encryption Key Manager Servers and the Encryption Key Manager CLI Client. This **keytool** command creates a new JCEKS keystore called EKMKey.jck and populates it with a certificate and private key with the alias of ekmcert. This certificate will be valid for five years. When this certificate expires, communications between Encryption Key Manager Servers and between the Encryption Key Manager CLI Client and Encryption Key Manager Server might no longer work. Remove the old expired certificate and create a new one as specified in this **keytool** command.

```
keytool -keystore EKMKeys.jck -storetype jceks -genkey -alias ekmcert -keyAlg RSA -keysize 2048 –validity 1825
```

When you issue this command, it prompts you for information it uses to create a distinguished name to put in the certificate. The prompts, with sample responses, look similar to these:

```
What is your first and last name? [Unknown]:  ekmcert
What is the name of your organizational unit?   [Unknown]:  EKM
What is the name of your organization? [Unknown]:  IBM
What is the name of your City or Locality?   [Unknown]: Austin
What is the name of your State or Province?   [Unknown]:  TX
What is the two-letter country code for this unit? [Unknown]:  US
Is CN=ekmcert, OU=EKM, O=IBM, L=Austin, ST=TX, C=US correct?(type "yes" or "no"):
```

Type yes and press Enter.

On i5/OS V5R3, Solaris, and HP-UX,, issuing the **keytool** command as shown above does not load the necessary IBM classes. Instead, use **java com.ibm.crypto.tools.KeyTool** with the same parameters, as in:

```
java com.ibm.crypto.tools.KeyTool -keystore EKMKeys.jck -storetype jceks -genkey
       -alias ekmcert -keyAlg RSA -keysize 2048 -validity 1825
```

This command prompts you for a password to access the keystore. This password must not be greater than 127 characters in length. Please note the keystore password entered here as it will be needed later when starting the Encryption Key Manager. When prompted for a *key* password, just press Enter. Do not type in a new or different password.

Step 4. Generate Encryption Keys

For LTO encryption, the Encryption Key Manager needs a number of symmetric keys to be pre-generated and stored in a keystore. This **keytool** command generates 32 256-bit AES keys and stores them in the keystore created in step 3. Run this command from the Encryption Key Manager directory to have the keystore file created in that directory. The resulting keys will have the names key000000000000000000 through key0000000000000001f.

```
keytool –keystore EKMKeys.jck –storetype jceks –genseckey –keyAlg aes –keysize 256 –aliasrange key00-1f
```

This command prompts you for a keystore password to access the keystore. Enter the desired password and press Enter. However, when prompted for a key password, just press Enter. Do not type in a new or different password. This will cause the key password to be the same as the keystore password. Please note the keystore password entered here as it will be needed later when starting the Encryption Key Manager.

On i5/OS V5R3, Solaris, and HP-UX,, issuing the **keytool** command as shown above does not load the necessary IBM classes. Instead, use **java com.ibm.crypto.tools.KeyTool** with the same parameters, as in:

```
java com.ibm.crypto.tools.KeyTool –keystore EKMKeys.jck –storetype jceks –genseckey –keyAlg aes
       –keysize 256 –aliasrange key00-1f
```

Step 5. Start the Encryption Key Manager Server

1. Change to the ekm1 directory. For example:

   On i5/OS

   ```
   cd /ekm/ekm1
   ```

   On Windows

   ```
   cd c:\ekm\ekm1
   ```

   On Unix platforms

   ```
   cd /var/ekm/ekm1
   ```

2. Enter this command:

   ```
   java com.ibm.keymanager.EKMLaunch KeyManagerConfig.properties
   ```

   This starts the Encryption Key Manager server.

Step 6. Start the Encryption Key Manager Command line Interface Client

1. From any command window or shell, enter:

   ```
   java com.ibm.keymanager.KMSAdminCmd ClientKeyManagerConfig.properties -i
   ```

2. At the # prompt, enter:

   ```
   login -ekmuser EKMAdmin -ekmpassword changeME
   ```

Once the CLI client is successfully logged into the key manager server, you can execute any CLI commands. Use the quit command to shut down the CLI client when you are finished. By default, the Encryption Key Manager server closes the communication socket with an unused client after ten minutes. Any attempt to enter a command after that will result in the client exiting. To specify a longer timeout period for the Encryption Key Manager server-client socket, modify theTransportListener.ssl.timeout property in the KeyManagerConfig.properties file.

Back up Your Keystore

Due to the critical nature of the keys in your keystore, it is highly recommended that you 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. There are many ways to backup this keystore information. Each keystore type has it own unique characteristics. These are discussed in more detail the *Encryption Key Manager Introduction, Planning, and User's Guide*, but these general guidelines apply to all:

- Keep a copy of all certificates loaded into the keystore (usually a PKCS12 format file).
- Use system backup capabilities to create a backup copy of the keystore information (be careful not to encrypt this copy using the encrypting tape drives as it would be impossible to decrypt it for recovery).
- Maintain a primary and secondary Encryption Key Manager and keystore copy (for backup as well as failover redundancy).
- If you are using a JCEKS keystore, simply copy the keystore file and store the clear (unencrypted) copy in a secure location such as a vault (be careful not to encrypt this copy using the encrypting tape drives as it would be impossible to decrypt it for recovery).

For More Information

See the following publication, along with the latest Encryption Key Manager support information, available at this website: http://www.ibm.com/support/docview.wss?&uid=ssg1S4000504.

- *IBM Encryption Key Manager component for the Java platform Introduction, Planning, and User's Guide*, GA76-0418.

Refer to the IBM System Storage Interoperation Center (SSIC) at http://www.ibm.com/systems/support/storage/config/ssic/displayesssearchwithoutjs.wss?start_over=yes for Open Systems configuration information.

Fourth edition

GA76-0420-03