

Administering

IBM

Contents

Administering	1	DS5000 management	101
Configuration settings	1	Administering devices, keys, and device associations	101
Specifying SSL or KMIP certificates	1	IBM Spectrum Scale file management	110
Specifying levels of audit information	3	Administering certificates and keys	111
Generating audit records in syslog format	5	Backup and restore	115
Specifying settings for debug information.	7	Backup and restore runtime requirements	115
Specifying key serving parameters	8	Backing up critical files	116
Determining the current port number.	11	Restoring a backup file	117
Specifying port and timeout settings	11	Installing Java Cryptography Extension unlimited strength jurisdiction policy files	119
Replication settings for clone and master servers	13	Starting and stopping the IBM Security Key Lifecycle Manager server on distributed systems	120
Administering groups, users, and roles	21	Deleting a backup file	121
Creating a group	21	Running backup and restore tasks on the command-line or REST interface	122
Assigning permissions.	23	Backup and restore operations for earlier versions of IBM Security Key Lifecycle Manager.	124
Creating a user in a group	26	Key loss prevention	153
Validating user tasks	28	Configuring automated backup script	154
Password policy for IBM Security Key Lifecycle Manager user.	29	KMIP objects management	156
Changing the password policy	30	Registering KMIP-compliant client devices	156
Changing a user password	30	Creating cryptographic objects for a client device	157
Changing IBM Security Key Lifecycle Manager user password	32	Modifying client device information.	158
Creating a device group	33	Deleting client device information	159
Creating a role for a new device group	34	Hardware Security Module usage in IBM Security Key Lifecycle Manager	159
Database administration	35	Configuring HSM parameters	161
Moving DB2 transaction log files for good performance	35	Configuration requirements to use HSM	162
DB2 password security issues on Windows systems.	36	LDAP configuration	162
DB2 password security issues on systems such as Linux or AIX	38	Integrating LDAP with IBM Security Key Lifecycle Manager.	164
Stopping the DB2 server	40	Post-LDAP configuration tasks to support LDAP integration	170
Changing the DB2 server host name	41	Exporting an SSL/KMIP server certificate	172
Changing an existing WebSphere Application Server host name	41	Copying a certificate between IBM Security Key Lifecycle Manager servers	173
Accepting pending devices	41	Changing the language of the browser interface	174
Moving devices between device groups	44	Notices	175
LTO tape drive management	47	Terms and conditions for product documentation	177
Guided steps to create key groups and drives ..	47	Trademarks	178
Managing keys, key groups, and drives	51	Index	179
3592 tape drive management	66		
Guided steps to create certificates and drives ..	67		
Administering certificates and devices	71		
DS8000 storage image management	85		
Guided steps to create storage images and image certificates.	85		
Administering storage images and image certificates.	89		

Administering

Administration is the set of tasks by which you prepare and then monitor the IBM Security Key Lifecycle Manager environment.

Administrative activities include tasks such as managing keys, certificates, and devices.

Configuration settings

IBM Security Key Lifecycle Manager provides a set of operations to change the IBM Security Key Lifecycle Manager configuration.

For example, you might change default values for the following processes:

- Port or timeout values for TCP and SSL communication
- Audit level that provides more log information
- Debug settings to generate debug logs
- Configuration settings for automated clone replication

Specifying SSL or KMIP certificates

You must specify the self-signed certificate to be used as server communication certificate. Alternatively, you might create requests for certificates and manually send the request to a certificate authority (CA) for signing. For example, you might use certificates to add protection to the communications between IBM Security Key Lifecycle Manager and a tape library. The generated certificate request files reside in the <SKLM_HOME> directory. For example, a generated certificate request might be a file such as *SKLM_HOME\080419154137-sslcert001.csr*.

About this task

You can use the SSL / KMIP for Key Serving page to specify the type of certificates that IBM Security Key Lifecycle Manager uses. Alternatively, you can use any of the following CLI commands or the REST interfaces:

- **tklmCertCreate** or **tklmCertGenRequest**
- **Certificate Generate Request REST Service** or **Create Certificate REST Service**

Your role must have a permission to the configure action to create an SSL or KMIP certificate.

Before you begin, determine:

- Whether you can use self-signed certificates during a phase in your project such as a test phase.
- The time interval that is needed to receive a CA-issued certificate after a request is sent. You must manually send a certificate request to the issuing authority.
- Whether your site requires partner certificates for use with business partners, vendors, or for disaster recovery purposes.
- The customary setting in days for a certificate validity interval.

Procedure

1. Navigate to the appropriate page or directory:
 - Graphical user interface:

Log on to the graphical user interface. Click **IBM Security Key Lifecycle Manager > Configuration > SSL/KMIP**.
 - Command-line interface:

In the `WAS_HOME/bin` directory, start a **wsadmin** session by using Jython. Log on to **wsadmin** with an authorized user ID, such as the SKLMAdmin user ID. For example, on Windows systems, navigate to the `drive:\Program Files (x86)\IBM\WebSphere\AppServer\bin` directory and type:

 - Windows systems:

```
wsadmin -username SKLMAdmin -password mypwd -lang jython
```
 - Systems such as AIX or Linux:

```
./wsadmin.sh -username SKLMAdmin -password mypwd -lang jython
```
 - REST interface:
 - Open a REST client.
2. Create one or more certificates or certificate requests:
 - Graphical user interface

Select whether to generate a self-signed certificate, or request a certificate from a third-party provider. There is also an option for the certificate to use an existing certificate from the keystore. Complete the required and optional fields, and then click **OK**.
 - Command-line interface

Type the **tklmCertCreate** command on one line. For example, to create a self-signed certificate, type:

```
print AdminTask.tklmCertCreate ('[-type selfsigned  
-alias sklmSSLCertificate -cn sklmssl -ou accounting -o myCompanyName  
-country US -keyStoreName defaultKeyStore  
-usage SSLSERVER -validity 999]')
```

You might alternatively request a certificate from a certificate authority. For example, type:

```
print AdminTask.tklmCertGenRequest('[-alias sklmSSLCertificate1  
-cn sklm -ou sales -o myCompanyName -locality myLocation  
-country US -validity 999 -keyStoreName defaultKeyStore  
-fileName mySSLCertRequest1.crt -usage SSLSERVER]')
```
 - REST interface
 - a. Obtain a unique user authentication identifier to access IBM Security Key Lifecycle Manager REST services. For more information about the authentication process, see Authentication process for REST services.
 - b. To invoke **Certificate Generate Request REST Service**, send the HTTP POST request. Pass the user authentication identifier that you obtained in Step a along with the request message as shown in the following example.

```
POST https://localhost:9080/SKLM/rest/v1/certificates  
Content-Type: application/json  
Accept : application/json  
Authorization: SKLMAuth authId=139aeh34567m  
Accept-Language : en  
{  
  "type": "selfsigned",  
  "alias": "sklmCertificate",  
  "cn": "sklm",  
  "ou": "sales",  
  "o": "myCompanyName",  
  "usage": "3592",  
  "country": "US",  
  "validity": "999",  
  "algorithm": "RSA"  
}
```

Send the following HTTP request for a certificate from a certificate authority:

```
POST https://localhost:9080/SKLM/rest/v1/certificates
Content-Type: application/json
Accept : application/json
Authorization: SKLMAuth authId=139aeh34567m
{"type":"certreq","alias":"sklmCert","cn":"sklm","ou":"sales","o":
"myCompanyName","usage":"3592","country":"US","validity":"999","fileName":
"myCertRequest1.crt","algorithm":"ECDSA"}
```

3. A success indicator varies, depending on the interface:

- Graphical user interface:

On the Success page, under Next Steps, click a related task that you want to carry out. If you create a self-signed certificate, you might restart the server and create a backup to ensure that you can restore this data.

- Command-line interface:

A completion message indicates success.

- REST interface:

The status code 200 OK indicates success.

What to do next

Go to the Welcome page and configure the drive types, and keys or certificates that your organization requires.

Specifying levels of audit information

You might change the default setting that IBM Security Key Lifecycle Manager uses to collect audit information.

About this task

You can use the Audit page to change information levels that are written to the audit log. Alternatively, you can use the following CLI commands or the REST interfaces to list or change the **Audit.event.types** property in the SKLMConfig.properties file:

- **tklmConfigGetEntry** and **tklmConfigUpdateEntry**
- **Get Single Config Property REST Service** and **Update Config Property REST Service**

Your role must have a permission to the configure action.

Procedure

1. Navigate to the appropriate page or directory:

- Graphical user interface:

Log on to the graphical user interface. Click **IBM Security Key Lifecycle Manager > Configuration > Audit and Debug**.

- Command-line interface:

In the *WAS_HOME/bin* directory, start a **wsadmin** session by using Jython. Log on to **wsadmin** with an authorized user ID, such as the SKLMAdmin user ID. For example, on Windows systems, navigate to the *drive:\Program Files (x86)\IBM\WebSphere\AppServer\bin* directory and type:

- Windows systems:

```
wsadmin -username SKLMAdmin -password mypwd -lang jython
```

- Systems such as AIX or Linux:

```
./wsadmin.sh -username SKLMAdmin -password mypwd -lang jython
```

- REST interface:
 - Open a REST client.
- 2. Change the value for the audit information level:
 - In the graphical user interface, select a low, medium, or high value for the Audit setting, then click **OK**.
 - Low** Stores minimal audit records.

Selecting **Low** sets the following property values in the SKLMConfig.properties file:

 - Audit.event.types = runtime, authorization, authorization_terminate, resource_management, key_management
 - Audit.event.outcome = failure
 - Medium (default)**

Stores an intermediate number of audit records.

Selecting **Medium** sets the following property values in the SKLMConfig.properties file:

 - Audit.event.types = runtime, authorization, authorization_terminate, resource_management, key_management
 - Audit.event.outcome = success, failure
 - High** Stores the maximum number of audit records.

Selecting **High** sets the following property values in the SKLMConfig.properties file:

 - Audit.event.types = all
 - Audit.event.outcome = success, failure
 - Command-line interface:
 - a. Type the **tklmConfigGetEntry** command on one line to get the current value of the target property in the SKLMConfig.properties file. For example, to determine which event types are included in the audit log, type on one line:


```
wsadmin>print AdminTask.tklmConfigGetEntry
('[-name Audit.event.types]')
```

An example response might be:

```
All
```
 - b. Specify the required change. For example, to limit the selection to two event types to store in the audit log, type on one line:


```
print AdminTask.tklmConfigUpdateEntry
('[-name Audit.event.types -value runtime,audit_management]')
```
- REST interface:
 - a. Obtain a unique user authentication identifier to access IBM Security Key Lifecycle Manager REST services. For more information about the authentication process, see Authentication process for REST services.
 - b. To invoke **Get Single Config Property REST Service**, send the HTTP GET request. Pass the user authentication identifier that you obtained in Step a along with the request message as shown in the following example.


```
GET https://localhost:9080/SKLM/rest/v1/configProperties/  
Audit.event.types  
Content-Type: application/json  
Accept : application/json  
Authorization: SKLMAuth userAuthId=139aeh34567m  
Accept-Language : en
```

Success response might be:

```
Status Code : 200 OK  
Content-Language: en  
{ "property": "Audit.event.types", "value": "all" }
```

- c. Specify the required change. For example, you can use **Update Config Property REST Service** to limit the selection to two event types to store in the audit log by sending the following HTTP request:

```
PUT https://localhost:9080/SKLM/rest/v1/configProperties  
Content-Type: application/json  
Accept: application/json  
Authorization: SKLMAuth authId=139aeh34567m  
Accept-Language: en  
{ "Audit.event.types": "runtime,audit_management" }
```

3. A success indicator varies, depending on the interface:

- Graphical user interface

On the Success page, under Next Steps, click a related task that you want to carry out.

- Command-line interface

A completion message indicates success.

- REST interface:

The status code 200 OK indicates success.

What to do next

You might rerun an operation that previously returned an error. Then, examine the audit log for more information. For detailed information about audit records, see the “Audit records on distributed systems” topic in IBM Security Key Lifecycle Manager documentation.

Generating audit records in syslog format

You can generate the audit records in syslog format and send them to a syslog server. Use the IBM Security Key Lifecycle Manager graphical user interface to configure for generating the audit records in syslog format.

About this task

The audit log messages are written to a configured local audit file in syslog format when:

- Syslog format is enabled for the audit messages.
- Syslog format is enabled, and syslog server host name and the port number are not specified.
- Syslog format is enabled, syslog server host name and port number are specified, but the server host name or port number is not reachable.

Procedure

1. Log on to the graphical user interface.
2. Click **IBM Security Key Lifecycle Manager > Configuration > Audit and Debug**.

3. Select **Use syslog format**.
4. Specify the server host name or IP address in **Syslog server host**.
5. Specify the port number on which the syslog server listens for requests in **Syslog server port**.
6. Select **Use SSL/TLS** if secure transfer of audit information to the syslog server by using the SSL/TLS transport protocol is needed.
7. Click **OK**.

What to do next

After you enabled syslog format for audit records with the requisite parameters, you must run the following steps only if you select **Use SSL/TLS**:

1. If the IBM Security Key Lifecycle Manager SSL server certificate is not already created, create the certificate. To create the certificate, you can use the SSL / KMIP for Key Serving page on graphical user interface, **Create Certificate REST Service**, or **tklmCertCreate** CLI command.
2. Export the IBM Security Key Lifecycle Manager SSL server certificate to a file. To export the certificate, you can use **Certificate Export REST Service** or **tklmCertExport** CLI command.

To export the server certificate, obtain the server certificate alias from Step 1 if the certificate is not already created. If the certificate is already created, from the graphical user interface, go to **Advanced Configuration > Server Certificates**. Alias is the **Certificates** column value for the certificate that is marked as In Use.

3. Obtain the syslog server certificate as a file, import it, and trust the syslog server certificate in IBM Security Key Lifecycle Manager server. Use **tklmCertImport** CLI command or **Certificate Import REST Service** to import the certificate by using SYSLOG usage.
4. Import the IBM Security Key Lifecycle Manager server certificate to syslog server. Use the certificate file that is created in Step 2.
5. Set the IBM Security Key Lifecycle Manager SSL server certificate alias in the configuration properties file.

Note: This step is not required if the IBM Security Key Lifecycle Manager SSL server certificate is created by using the graphical user interface.

For example:

Command-line interface

```
print AdminTask.tklmConfigUpdateEntry('[-name config.keystore.ssl.
certalias -value <alias of the server certificate that is
created in Step 1>']')
```

REST interface

```
PUT https://localhost:9080/SKLM/rest/v1/configProperties
Content-Type: application/json
Accept : application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
Accept-Language : en
{ "config.keystore.ssl.certalias" : "<alias of the server
certificate that is created in Step 1>" }
```

6. Restart IBM Security Key Lifecycle Manager server.

Specifying settings for debug information

You can change the default setting that IBM Security Key Lifecycle Manager uses to collect debug information. Debug log files provide additional information to analyze and troubleshoot IBM Security Key Lifecycle Manager problems.

About this task

You can use the Debug section of the Audit page to specify settings for generating debug information. Alternatively, you can use the following CLI commands or the REST interfaces to list or change the **debug** property in the SKLMConfig.properties file:

- **tklmConfigGetEntry** and **tklmConfigUpdateEntry**
- **Get Single Config Property REST Service** and **Update Config Property REST Service**

Your role must have a permission to the configure action.

Note: Enabling debug logging might affect IBM Security Key Lifecycle Manager performance. Enable this option only under the guidance of your IBM support representative.

Procedure

1. Navigate to the appropriate page or directory:
 - Graphical user interface:
Log on to the graphical user interface. Click **IBM Security Key Lifecycle Manager > Configuration > Audit and Debug**.
 - Command-line interface:
In the *WAS_HOME/bin* directory, start a **wsadmin** session by using Jython. Log on to **wsadmin** with an authorized user ID, such as the SKLMAdmin user ID. For example, on Windows systems, navigate to the *drive:\Program Files (x86)\IBM\WebSphere\AppServer\bin* directory and type:
 - Windows systems:

```
wsadmin -username SKLMAdmin -password mypwd -lang jython
```
 - Systems such as AIX or Linux:

```
./wsadmin.sh -username SKLMAdmin -password mypwd -lang jython
```
 - REST interface:
 - Open a REST client.
2. Change the settings to generate debug information:
 - In the graphical user interface:
 - a. Select **Enable debug** to set the following property values in the SKLMConfig.properties file:

```
debug=all
```
 - b. Click **OK**.
 - Command-line interface:
 - a. Type the **tklmConfigGetEntry** command on one line to get the current value of the target property in the SKLMConfig.properties file. For example, to determine the value of debug, type on one line:

```
wsadmin>print AdminTask.tklmConfigGetEntry  
('[-name debug]')
```


An example response might be:

```
none
```

- b. Specify a new value for the property. For example, to specify the value `all` for generating debug logs, type on one line:

```
print AdminTask.tklmConfigUpdateEntry
  ('[-name debug -value all]')
```
 - REST interface:
 - a. Obtain a unique user authentication identifier to access IBM Security Key Lifecycle Manager REST services. For more information about the authentication process, see [Authentication process for REST services](#).
 - b. To invoke **Get Single Config Property REST Service**, send the HTTP GET request. Pass the user authentication identifier that you obtained in Step a along with the request message as shown in the following example.

```
GET https://localhost:9080/SKLM/rest/v1/configProperties/debug
Content-Type: application/json
Accept : application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
Accept-Language : en
```

Success response might be:

```
Status Code : 200 OK
Content-Language: en
{"property":"debug","value":"none"}
```
 - c. Specify a new value for the property. Then, send the following HTTP request:

```
PUT https://localhost:9080/SKLM/rest/v1/configProperties
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth authId=139aeh34567m
Accept-Language: en
{ "debug": "all"}
```
3. A success indicator varies, depending on the interface:
 - Graphical user interface

On the Success page, under Next Steps, click a related task that you want to run.
 - Command-line interface

A completion message indicates success.
 - REST interface:

The status code 200 OK indicates success.

Specifying key serving parameters

You might change the default certificate settings that IBM Security Key Lifecycle Manager provides.

About this task

Use the Key Serving Parameters page to change certificate settings. Alternatively, you can use the following CLI commands or the REST interfaces to list or change the appropriate properties in the `SKLMConfig.properties` file:

- **tklmConfigGetEntry** and **tklmConfigUpdateEntry**
- **Get Single Config Property REST Service** and **Update Config Property REST Service**

Your role must have a permission to the configure action.

Before you begin, determine whether:

- To carry out certificate date validation before a key is served. Validation confirms that the certificate is valid, and is not expired.
- To identify certificates by using the subject key identifier that is stored in the certificate.

Procedure

1. Navigate to the appropriate page or directory:

- Graphical user interface:

Log on to the graphical user interface. Click **IBM Security Key Lifecycle Manager > Configuration > Key Serving Parameters**.

- Command-line interface:

In the *WAS_HOME/bin* directory, start a **wsadmin** session by using Jython. Log on to **wsadmin** with an authorized user ID, such as the SKLMAdmin user ID. For example, on Windows systems, navigate to the *drive:\Program Files (x86)\IBM\WebSphere\AppServer\bin* directory and type:

- Windows systems:

```
wsadmin -username SKLMAdmin -password mypwd -lang jython
```

- Systems such as AIX or Linux:

```
./wsadmin.sh -username SKLMAdmin -password mypwd -lang jython
```

- REST interface:

- Open a REST client.

2. Change the value for one or more certificate settings:

- In the graphical user interface, change one or more of the following settings, and then click **OK**:

Do not use expired certificates for write requests or data writes.

Before you serve a key, validates that the expiration date is not passed for the certificate or certificates that wraps this key. Expired certificates are used only for read requests. When this setting is enabled, expired certificates are not used for write requests. Selecting this check box changes the value of the **cert.validate** property to true in the *SKLMConfig.properties* file.

Keep pending client device communication certificates.

Keep communication certificates from client devices pending until you accept the certificates for use in secure communication between the device and the IBM Security Key Lifecycle Manager server. If you disable this setting, you must manually import client device communication certificates. This configuration parameter is associated with the value of the **enableClientCertPush** property from client devices pending in the *SKLMConfig.properties* file.

Identify certificates by certificate name.

Identify certificates by using the certificate name that is stored in the certificate, rather than using a subject key identifier. You specify the certificate name when you create a certificate. This function is used when decrypting data that was written to a device.

When disabled, the Subject Key Identifier is used to determine the certificate to be used when reading data on a cartridge or other device. This configuration parameter is associated with the value of the **useSKIdefaultLabels** property in the *SKLMConfig.properties* file.

- Command-line interface:

- a. Type the **tklmConfigGetEntry** command on one line to get the current value of the target property in the SKLMConfig.properties file. For example, type:

```
wsadmin>print AdminTask.tklmConfigGetEntry
('[-name zOSCompatibility]')
```

An example response might be:

```
False
```

- b. Specify the required change. For example, to change the value of the **zOSCompatibility** property to true, type on one line:

```
print AdminTask.tklmConfigUpdateEntry
('[-name zOSCompatibility -value true]')
```

- REST interface:

- a. Obtain a unique user authentication identifier to access IBM Security Key Lifecycle Manager REST services. For more information about the authentication process, see Authentication process for REST services.
- b. To invoke **Get Single Config Property REST Service**, send the HTTP GET request. Pass the user authentication identifier that you obtained in Step a along with the request message as shown in the following example.

Service request

```
GET https://localhost:9080/SKLM/rest/v1/configProperties/
zOSCompatibility
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
Accept-Language: en
```

Success response

```
Status Code : 200 OK
Content-Language: en
{"zOSCompatibility" : "False"}
```

- c. Specify the required change. For example, you can send the following service request to change the value of the **zOSCompatibility** property to true:

```
PUT https://localhost:9080/SKLM/rest/v1/configProperties
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth authId=139aeh34567m
Accept-Language: en
{ "zOSCompatibility": "true"}
```

3. A success indicator varies, depending on the interface:

- Graphical user interface:

On the Success page, under Next Steps, click a related task that you want to carry out.

- Command-line interface:

A completion message indicates success.

- REST interface:

The status code 200 OK indicates success.

What to do next

Changes to certificate settings occur dynamically. Next, you might create the necessary certificates and associate them with specific devices.

Determining the current port number

After IBM Security Key Lifecycle Manager server installation, you might want to determine the secure port numbers for the IBM Security Key Lifecycle Manager server and the WebSphere Integrated Solutions Console.

About this task

The value of the port numbers is specified by the **WC_defaulthost_secure** or the **WC_adminhost_secure** property in the *WAS_HOME/profiles/KLMProfile/properties/portdef.props* file. For example, the file might specify these values:

```
WC_defaulthost_secure=9080
WC_adminhost_secure=9083
```

The **WC_defaulthost_secure** property value corresponds to the IBM Security Key Lifecycle Manager server secure port and the **WC_adminhost_secure** property value corresponds to the WebSphere Integrated Solutions Console secure port.

Specifying port and timeout settings

You might change the default port and timeout settings that IBM Security Key Lifecycle Manager provides.

About this task

You can use the Key Serving Parameters page to change port and timeout settings. Alternatively, you can use the following CLI commands or the REST services to list and change the appropriate properties in the *SKLMConfig.properties* file:

- **tklmConfigGetEntry** and **tklmConfigUpdateEntry**
- **Get Single Config Property REST Service** and **Update Config Property REST Service**

Before you begin, determine whether there are port or timeout conflicts at your site that prevent from using the IBM Security Key Lifecycle Manager default values. Your role must have a permission to the configure action.

Procedure

1. Navigate to the appropriate page or directory:

- Graphical user interface:

Log on to the graphical user interface. Click **IBM Security Key Lifecycle Manager > Configuration > Key Serving Ports**.

- Command-line interface:

In the *WAS_HOME/bin* directory, start a **wsadmin** session by using Jython. Log on to **wsadmin** with an authorized user ID, such as the SKLMAdmin user ID. For example, on Windows systems, navigate to the *drive:\Program Files (x86)\IBM\WebSphere\AppServer\bin* directory and type:

- Windows systems:

```
wsadmin -username SKLMAdmin -password mypwd -lang jython
```

- Systems such as AIX or Linux:

```
./wsadmin.sh -username SKLMAdmin -password mypwd -lang jython
```

- REST interface:

- Open a REST client.

2. Change the value for the port or timeout settings:

- In the graphical user interface, change one or more of these settings, and then click **OK**:

TCP port

IBM Security Key Lifecycle Manager uses default port 3801. Values can range from 1 to 65535. The value that you set also changes the value of the **TransportListener.tcp.port** property in the SKLMConfig.properties file. You must ensure that the port is not already in use by another application.

TCP timeout (in minutes)

IBM Security Key Lifecycle Manager uses a default timeout value of 10 minutes. Values can range from 1 to 120. The value that you set also changes the value of the **TransportListener.tcp.timeout** property in the SKLMConfig.properties file.

SSL port

IBM Security Key Lifecycle Manager uses default port 441. Values can range from 1 to 65535. The value that you set also changes the value of the **TransportListener.ssl.port** property in the SKLMConfig.properties file.

SSL timeout (in minutes)

IBM Security Key Lifecycle Manager uses a default timeout value of 10 minutes. Values can range from 1 to 120. This configuration parameter is associated with the value of the **TransportListener.ssl.timeout** property in the SKLMConfig.properties file.

KMIP SSL port

KMIP uses default port 5696. Values can range from 1 to 65535. This configuration parameter is associated with the value of the **KMIPListener.ssl.port** property in the SKLMConfig.properties file.

- Command-line interface:
 - a. Type the **tklmConfigGetEntry** command on one line to get the current value of the target property in the SKLMConfig.properties file. For example, type on one line:


```
wsadmin>print AdminTask.tklmConfigGetEntry
('[-name TransportListener.tcp.port]')
```

 An example response might be:


```
3801
```
 - b. Specify the required change. For example, to specify a different TCP port number, type on one line:


```
print AdminTask.tklmConfigUpdateEntry
('[-name TransportListener.tcp.port -value 3802]')
```
- REST interface:
 - a. Obtain a unique user authentication identifier to access IBM Security Key Lifecycle Manager REST services. For more information about the authentication process, see Authentication process for REST services.
 - b. To invoke **Get Single Config Property REST Service**, send the HTTP GET request. Pass the user authentication identifier that you obtained in Step a along with the request message as shown in the following example.

Service request

```
GET https://localhost:9080/SKLM/rest/v1/configProperties/  
TransportListener.tcp.port  
Content-Type: application/json  
Accept: application/json  
Authorization: SKLMAuth userAuthId=139aeh34567m  
Accept-Language: en
```

Success response

```
Status Code : 200 OK  
Content-Language: en  
{"TransportListener.tcp.port" : "3801"}
```

- c. Specify the required change. For example, to specify a different TCP port number, send the following service request:

```
PUT https://localhost:9080/SKLM/rest/v1/configProperties  
Content-Type: application/json  
Accept: application/json  
Authorization: SKLMAuth authId=139aeh34567m  
Accept-Language: en  
{"TransportListener.tcp.port": "3802"}
```

3. A success indicator varies, depending on the interface:

- Graphical user interface:
An update page displays the information that you entered.
- Command-line interface:
A completion message indicates success.
- REST interface:
The status code 200 OK indicates success.

What to do next

To put a change such as a port number into effect, restart the IBM Security Key Lifecycle Manager server.

Replication settings for clone and master servers

To set up the IBM Security Key Lifecycle Manager automated clone replication process, you must configure the replication parameters for master and clone servers.

IBM Security Key Lifecycle Manager provides a set of operations to replicate current active files and data across systems. This replication enables cloning of IBM Security Key Lifecycle Manager environments to multiple servers in a manner that is independent of operating systems and directory structure of the server. For example, you can replicate data from a master server on a Windows system to a clone server on a Linux system.

Master server configuration

Master server is the primary system that is being replicated. Replication process is triggered only when the new keys are added to the master server. You can replicate the master server with a maximum of 20 clone servers. Each clone server is identified through an IP address or host name, and a port number. The server uses properties in the `ReplicationSKLMConfig.properties` file to control the replication process. For more information about replication configuration file, see Replication configuration files.

You can also use the IBM Security Key Lifecycle Manager replication program to schedule automatic backup operation. You must configure properties only for the master server to back up data at regular intervals.

Clone server configuration

The replication process enables cloning of IBM Security Key Lifecycle Manager environments from master server to multiple clone servers. The clone server uses properties in the `ReplicationSKLMConfig.properties` file to control the replication process. When the replication process is triggered, the following data is replicated to the clone server:

- Data in the IBM Security Key Lifecycle Manager database tables
- Truststore and keystore with the master key
- IBM Security Key Lifecycle Manager configuration files

Specifying replication parameters for a master server

You can change default settings of master server for communication with the clone server to replicate IBM Security Key Lifecycle Manager data.

About this task

Use the Automated Clone Replication Configuration page to configure replication settings. Alternatively, you can use the following CLI commands or the REST interfaces to change the appropriate properties in the `ReplicationSKLMConfig.properties` configuration file:

- **`tklmReplicationConfigGetEntry`** and **`tklmReplicationConfigUpdateEntry`**
- **Get Single Replication Config Properties REST Service** and **Update Replication Config Property REST Service**

Note: Data is replicated to the clone server only when the new keys are added to the master server.

Procedure

1. Navigate to the appropriate page or directory:
 - Graphical user interface:
 - a. Log on to the graphical user interface.
 - b. Click **IBM Security Key Lifecycle Manager > Configuration > Replication**.
 - Command-line interface:

In the `WAS_HOME/bin` directory, start a **wsadmin** session by using Jython. Log on to **wsadmin** with an authorized user ID, such as the SKLMAdmin user ID. For example, on Windows systems, navigate to the `drive:\Program Files (x86)\IBM\WebSphere\AppServer\bin` directory and type:

 - Windows systems:

```
wsadmin -username SKLMAdmin -password mypwd -lang jython
```
 - Systems such as AIX or Linux:

```
./wsadmin.sh -username SKLMAdmin -password mypwd -lang jython
```
 - REST interface:
 - Open a REST client.
2. Change the value for one or more settings of the master server:
 - In the graphical user interface:

- a. Select **Master**.
- b. Specify the appropriate settings:

Basic Properties

Certificate from keystore	Select a certificate from the list. The SSL/TLS certificate must exist on the master and all clone systems that you configure for replication.
Replication backup encryption passphrase	Encryption password for the backup file to ensure data security. You need the same password to decrypt and restore the file.
Confirm replication backup encryption passphrase	Specify the same password again to verify correct entry.
Master listen port	Port number for communication when unserialized or delayed replications take place. Default master listen port is 1111.
Clone -1 IP or Host name	IP address or host name of the clone servers. You can replicate only 1 master server with a maximum of 20 clone servers. Click the Add Clone link to configure replication settings for multiple clones.
Clone -1 Port	Port number for sending backup files to the clone servers. Each clone server is identified through a port number. Default port number for clone server is 2222.

Advanced Properties

Replication backup destination directory	Location to store the backup files. Default destination directory is <WAS_HOME>\products\sklm\restores.
Maximum number of replication files to keep before rollover	Maximum number of replication files that you want to keep. The value must be a positive integer between 2 - 10. When the number of files exceed the specified limit, the oldest file is deleted.
Replication frequency (in hours)	Frequency to check whether the backup operation is necessary. Default value is set to 1 hour. This parameter is ignored if the value for Daily Start Replication Time is set.
Daily replication time (in HH:MM format)	Time in HH:MM format to run the replication task every day.
Replication log file name	Name and location for the replication log file. Default value for this parameter is <WAS_HOME>\products\sklm\logs\replication.
Maximum log file size (in KB)	Maximum size of a log file before rollover occurs. Default value is 1000 KB (kilobytes). When the file reaches the maximum size, a new log file is created.
Maximum number of log files to keep	Maximum number of log files that you want to keep. By default, IBM Security Key Lifecycle Manager keeps the last 3 log files. When the number of files exceed the specified limit, the oldest file is deleted.

- c. Click **OK**.
- Command-line interface:
 - a. Type the **tklmReplicationConfigGetEntry** command on one line to get the current value of the target property in the ReplicationSKLMConfig.properties file. For example, type

```
wsadmin>print AdminTask.tklmReplicationConfigGetEntry  
  (['-name replication.role'])
```

An example response might be:

```
none
```

- b. Specify the changes. For example, to change the value of the **replication.role** property to master, type on one line.

```
print AdminTask.tklmReplicationConfigUpdateEntry  
  (['-name replication.role -value master'])
```

- REST interface:

- a. Obtain a unique user authentication identifier to access IBM Security Key Lifecycle Manager REST services. For more information about the authentication process, see Authentication process for REST services.
- b. To invoke **Get Single Config Property REST Service**, send the HTTP GET request. Pass the user authentication identifier that you obtained in Step a along with the request message as shown in the following example.

Service request

```
GET https://localhost:9080/SKLM/rest/v1/configProperties/  
replication.role  
Content-Type: application/json  
Accept: application/json  
Authorization: SKLMAuth userAuthId=139aeh34567m  
Accept-Language: en
```

Success response

```
Status Code : 200 OK  
Content-Language: en  
{ "replication.role" : "none" }
```

- c. Specify the changes. For example, you can use **Update Replication Config Property REST Service** to send the following service request to change the value of the **replication.role** property.

```
PUT https://localhost:9080/SKLM/rest/v1/configProperties  
Content-Type: application/json  
Accept: application/json  
Authorization: SKLMAuth authId=139aeh34567m  
Accept-Language: en  
{ "replication.role": "master" }
```

3. A success indicator varies, depending on the interface:

- Graphical user interface:

Updated configuration information is displayed on the Automated Clone Replication Configuration page for master server.

- Command-line interface:

A completion message indicates success.

- Rest interface:

The status code 200 OK indicates success.

What to do next

You might want to change the settings for clone servers to receive backup files from the master server.

Specifying replication parameters for a clone server

You can change default settings of the clone server to automatically receive backup files from master server when the new keys are added to the server.

About this task

Use the Automated Clone Replication Configuration page to change replication settings. Alternatively, you can use the following CLI commands or the REST interfaces to list or change the appropriate properties in the `ReplicationSKLMConfig.properties` configuration file:

- `tklmReplicationConfigGetEntry` and `tklmReplicationConfigUpdateEntry`
- **Get Single Replication Config Properties REST Service** and **Update Replication Config Property REST Service**

Procedure

1. Navigate to the appropriate page or directory:

- Graphical user interface:
 - a. Log on to the graphical user interface.
 - b. Click **IBM Security Key Lifecycle Manager > Configuration > Replication**.

- Command-line interface:

In the `WAS_HOME/bin` directory, start a **wsadmin** session by using Jython. Log on to **wsadmin** with an authorized user ID, such as the SKLMAdmin user ID. For example, on Windows systems, navigate to the `drive:\Program Files (x86)\IBM\WebSphere\AppServer\bin` directory and type:

- Windows systems:

```
wsadmin -username SKLMAdmin -password mypwd -lang jython
```

- Systems such as AIX or Linux:

```
./wsadmin.sh -username SKLMAdmin -password mypwd -lang jython
```

- REST interface:
 - Open a REST client.

2. Change the value for one or more settings of the clone server:

- In the graphical user interface:
 - a. Select **Clone**.
 - b. Specify the appropriate settings:

Basic Properties

Clone listen port	Port number that the clone server must listen on to receive backup files. Default port number is 2222.
Master listen port	Port number for communication when unserialized or delayed replications take place. Default master listen port is 1111.

Advanced Properties

Number of retries incase of restore failure	Maximum number of retries that are allowed after the first restore operation is failed. The value must be a positive integer between 0 - 2.
Replication log file name	Name and location for the replication log file. Default value for this parameter is <code><WAS_HOME>\products\sklm\logs\replication</code> .
Maximum log file size (in KB)	Maximum size of a log file before rollover occurs. Default value is 1000 KB (kilobytes). When the file reaches the maximum size, a new log file is created.

Maximum number of log files to keep	Maximum number of log files that you want to keep. By default, IBM Security Key Lifecycle Manager keeps the last 3 log files. When the number of files exceed the specified limit, the oldest file is deleted.
--	--

- c. Click **OK**.
 - Command-line interface:
 - a. Type the **tklmReplicationConfigGetEntry** command on one line to get the current value of the target property in the ReplicationSKLMConfig.properties file. For example, type


```
wsadmin>print AdminTask.tklmReplicationConfigGetEntry
('[-name replication.role]')
```

 An example response might be:


```
none
```
 - b. Specify the changes. For example, to change the value of the **replication.role** property to clone, type on one line.


```
print AdminTask.tklmReplicationConfigUpdateEntry
('[-name replication.role -value clone]')
```
 - REST interface:
 - a. Obtain a unique user authentication identifier to access IBM Security Key Lifecycle Manager REST services. For more information about the authentication process, see Authentication process for REST services.
 - b. To invoke **Get Single Config Property REST Service**, send the HTTP GET request. Pass the user authentication identifier that you obtained in Step a along with the request message as shown in the following example.

Service request

```
GET https://localhost:9080/SKLM/rest/v1/configProperties/
replication.role
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
Accept-Language: en
```

Success response

```
Status Code : 200 OK
Content-Language: en
{"replication.role" : "none"}
```
 - c. Specify the changes. For example, you can use **Update Replication Config Property REST Service** to send the following service request to change the value of the **replication.role** property.


```
PUT https://localhost:9080/SKLM/rest/v1/configProperties
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth authId=139aeh34567m
Accept-Language: en
{ "replication.role": "clone"}
```
3. A success indicator varies, depending on the interface:
 - Updated configuration information is displayed on the Automated Clone Replication Configuration page for clone server.
 - Command-line interface:

A completion message indicates success.
 - Rest interface:

The status code 200 OK indicates success.

What to do next

You might want to change the settings for other clone servers. You can replicate IBM Security Key Lifecycle Manager data from a primary master server up to 20 secondary clone servers.

Scheduling automatic backup operation

You can configure replication settings to automatically run the backup operation to ensure that IBM Security Key Lifecycle Manager critical data is backed up at regular intervals.

About this task

You can use the IBM Security Key Lifecycle Manager replication program to schedule automatic backup operation. You must configure properties only for the master server to back up data at regular intervals.

Use the Automated Clone Replication Configuration page to configure settings for master server. Alternatively, you can use the following CLI commands or the REST interfaces to change the appropriate properties in the `ReplicationSKLMConfig.properties` configuration file:

- `tklmReplicationConfigGetEntry` and `tklmReplicationConfigUpdateEntry`
- **Get Single Replication Config Properties REST Service** and **Update Replication Config Property REST Service**

Procedure

1. Navigate to the appropriate page or directory:
 - Graphical user interface:
 - a. Log on to the graphical user interface.
 - b. Click **IBM Security Key Lifecycle Manager > Configuration > Replication**.
 - Command-line interface:

In the `WAS_HOME/bin` directory, start a **wsadmin** session by using Jython. Log on to **wsadmin** with an authorized user ID, such as the SKLMAdmin user ID. For example, on Windows systems, navigate to the `drive:\Program Files (x86)\IBM\WebSphere\AppServer\bin` directory and type:

 - Windows systems:

```
wsadmin -username SKLMAdmin -password mypwd -lang jython
```
 - Systems such as AIX or Linux:

```
./wsadmin.sh -username SKLMAdmin -password mypwd -lang jython
```
 - REST interface:
 - Open a REST client.
2. Change the value for one or more settings of the master server:
 - In the graphical user interface:
 - Select **Master**.
 - Configure the settings:

Basic Properties

Certificate from keystore	Select a certificate from the list. The SSL/TLS certificate must exist on the master and all clone systems that you configure for replication.
---------------------------	--

Replication backup encryption passphrase	Encryption password for the backup file to ensure data security. You need the same password to decrypt and restore the file.
Confirm replication backup encryption passphrase	Specify the same password again to verify correct entry.
Master listen port	Port number for communication when unserialized or delayed replications take place. Default master listen port is 1111.

Advanced Properties

Replication backup destination directory	Location to store the backup files. Default destination directory is <WAS_HOME>\products\sklm\restores.
Maximum number of replication files to keep before rollover	Maximum number of replication files that you want to keep. The value must be a positive integer between 2 - 10. When the number of files exceed the specified limit, the oldest file is deleted.
Replication frequency (in hours)	Frequency to check whether the backup operation is necessary. Default value is set to 1 hour. This parameter is ignored if the value for Daily Start Replication Time is set.
Daily replication time (in HH:MM format)	Time in HH:MM format to run the replication task every day.
Replication log file name	Name and location for the replication log file. Default value for this parameter is <WAS_HOME>\products\sklm\logs\replication.
Maximum log file size (in KB)	Maximum size of a log file before rollover occurs. Default value is 1000 KB (kilobytes). When the file reaches the maximum size, a new log file is created.
Maximum number of log files to keep	Maximum number of log files that you want to keep. By default, IBM Security Key Lifecycle Manager keeps the last 3 log files. When the number of files exceed the specified limit, the oldest file is deleted.

- Click **OK**.
- Command-line interface:
 - a. Type the **tklmReplicationConfigGetEntry** command on one line to get the current value of the target property in the ReplicationSKLMConfig.properties file. For example, type:


```
wsadmin>print AdminTask.tklmReplicationConfigGetEntry
('[-name replication.role]')
```

 An example response might be:


```
none
```
 - b. Specify the changes. For example, to change the value of the **replication.role** property to master, type on one line.


```
print AdminTask.tklmReplicationConfigUpdateEntry
('[-name replication.role -value master]')
```
- REST interface:
 - a. Obtain a unique user authentication identifier to access IBM Security Key Lifecycle Manager REST services. For more information about the authentication process, see Authentication process for REST services.

- b. To invoke **Get Single Config Property REST Service**, send the HTTP GET request. Pass the user authentication identifier that you obtained in Step a along with the request message as shown in the following example.

Service request

```
GET https://localhost:9080/SKLM/rest/v1/configProperties/
replication.role
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
Accept-Language: en
```

Success response

```
Status Code : 200 OK
Content-Language: en
{"replication.role" : "none"}
```

- c. Specify the changes. For example, you can use **Update Replication Config Property REST Service** to send the following service request to change the value of the **replication.role** property.

```
PUT https://localhost:9080/SKLM/rest/v1/configProperties
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth authId=139aeh34567m
Accept-Language: en
{ "replication.role": "master" }
```

3. A success indicator varies, depending on the interface:

- Graphical user interface:

Updated configuration information is displayed on the Automated Clone Replication Configuration page of master server.

- Command-line interface:

A completion message indicates success.

- Rest interface:

The status code 200 OK indicates success.

Administering groups, users, and roles

You can limit the range of activities that administrators can carry out in your organization.

For long-term efficiency, consider creating a group and then assigning roles and users to the group, rather than assigning roles directly to an individual user. You gain ease in changing roles for persons with similar duties, and avoid rework if a user is assigned to another department.

For example, you might specify this range of activities:

- No access is available for some roles. For example, your organization might want to separate the duties that back up and restore files.
- Some tasks are hidden on WebSphere® Integrated Solutions Console.
- Administration can occur only to LTO tape drives.

Creating a group

You can create a group that you intend to use to specify limits for some system administrators. You must model the group after the predefined LTO groups.

About this task

This task uses the WASAdmin user ID on the WebSphere Integrated Solutions Console to create an administrative group.

Note: To access IBM Security Key Lifecycle Manager graphical user interface or command-line interface, the user must be assigned to this group:
k1mGUICLIAccessGroup

For more information about the commands that create groups and users, see the IBM WebSphere Application Server documentation (http://www-01.ibm.com/support/knowledgecenter/SSAW57_8.5.5/com.ibm.websphere.nd.iseries.doc/ae/rxml_atwimmgt.html).

You can use WebSphere Integrated Solutions Console to create child groups with different permissions within a parent group. However, IBM Security Key Lifecycle Manager recognizes the permissions of only the parent group, not the permissions of its child groups.

Procedure

1. Log on to WebSphere Integrated Solutions Console (<https://localhost:9083/ibm/console/logon.jsp>).
 - Graphical user interface:
 - a. On the browser Welcome page, type the user ID WASAdmin and the password for this administrator.
 - b. In the navigation tree, click **Users and Groups > Manage Groups**.
 - Command-line interface:

In the *WAS_HOME/bin* directory, start a **wsadmin** session by using Jython. Log on to **wsadmin** with an authorized user ID, such as the WASAdmin user ID. For example, on Windows systems, navigate to the *drive:\Program Files (x86)\IBM\WebSphere\AppServer\bin* directory and type:

 - Windows systems:

```
wsadmin -username WASAdmin -password wasadminpw -lang jython
```
 - Systems such as AIX or Linux:

```
./wsadmin.sh -username WASAdmin -password wasadminpw -lang jython
```
2. Create a group:
 - Graphical user interface:
 - a. On the Manage Groups page, click **Create**.
 - b. In the **Group name** field, specify the group name. For example, type DS5000Admin.
 - c. In the **Description** field, specify more information about the group that you want to create.
 - d. Click **Create**.
 - Command-line interface:
 - a. Create an authorization group.
 - b. Create a group.

Type createGroup and specify the required values to create a group. For example, by using Jython, type:

```
print AdminTask.createGroup  
  ('[-cn DS5000Admin -description DS5000_LocalAdmins]')
```

where:

-cn Required (string). Specifies the common name for the group that you want to create. This parameter maps to the **cn** property in virtual member manager.

-description

Optional (string). Specifies more information about the group that you want to create.

3. Save your work.

- Graphical user interface:

Confirm completion of your task, by using the prompt that the graphical user interface provides.

- Command-line interface:

Save your configuration. For example, by using Jython, type:

```
print AdminConfig.save()
```

What to do next

Next, assign one or more permissions or roles to the group.

Assigning permissions

You can map an administrative group to a limited set of permissions.

About this task

This task uses the WASAdmin user ID on the WebSphere Integrated Solutions Console to map a group to a limited set of actions to administer DS5000 storage servers.

For more information about the commands that map groups to roles, see the IBM WebSphere Application Server documentation (http://www-01.ibm.com/support/knowledgecenter/SSAW57_8.5.5/com.ibm.websphere.nd.doc/ae/rxml_atauthorizationgroup.html).

Procedure

1. Log on to WebSphere Integrated Solutions Console.

- Graphical user interface:

a. On the browser Welcome page, type a user ID of WASAdmin and a password value, such as wasadminpw.

b. In the graphical user interface, click **Users and Groups > Administrative group roles**.

c. Click **Add**.

- Command-line interface:

In the *WAS_HOME/bin* directory, start a **wsadmin** session by using Jython. Log on to **wsadmin** with an authorized user ID, such as the WASAdmin user ID. For example, on Windows systems, navigate to the *drive:\Program Files (x86)\IBM\WebSphere\AppServer\bin* directory and type:

– Windows systems:

```
wsadmin -username WASAdmin -password wasadminpw -lang jython
```

– Systems such as AIX or Linux:

```
./wsadmin.sh -username WASAdmin -password wasadminpw -lang jython
```

2. Map a limited set of roles to the group.

- Graphical user interface:
 - a. In the Administrative group roles page, in the General Properties section, click **Enter group name**. In the **Group name** field, type the name of the group. For example, type DS5000Admin.
 - b. In the General Properties section, select the required subset of roles from the **Roles** list. For example, take these steps:
 - Block access to some roles. For example, your organization might want to separate the duties that restore files. In that case, do not select the **klmRestore** item in the list.
 - Determine whether you want to hide other tasks on the WebSphere Integrated Solutions Console. If you do hide tasks, select **suppressmonitor** as a role.
 - Limit administration only to DS5000 storage servers. For example, select **DS5000**.
Alternatively, if your task defines administrative activities for a new device group such as myDS5000, you might select myDS5000, which you previously created.
 - Press the Ctrl key and select roles that apply to IBM Security Key Lifecycle Manager:
 - klmBackup**
Create and delete a backup of data.
 - klmRestore**
Restore a previous backup copy of data.
 - klmConfigure**
Read or change properties, or act on certificates.
 - klmAudit**
View audit data.
 - klmView**
View objects.
 - klmCreate**
Create objects.
 - klmModify**
Modify objects.
 - klmDelete**
Delete objects.
 - klmGet**
Export a key or certificate.
 - suppressmonitor**
Hide other tasks on the WebSphere Integrated Solutions Console.
 - DS5000**
Allows actions on DS5000 storage servers.
 - c. Click **OK**.
 - d. Click **Save** to save your changes directly to the master configuration.
- Command-line interface:

Type `mapGroupsToAdminRole` and specify the required values to map the group to a specific administrative role. For example, by using Jython to specify more than one role to a group, type a sequence of commands, pressing **Enter** after each command.

- Specify the first role for the group:

```
print AdminTask.mapGroupsToAdminRole('[-roleName suppressmonitor
-groupids DS5000Admin]')
```

- Specify the next role for the group:

```
print AdminTask.mapGroupsToAdminRole('[-roleName klmConfigure
-groupids DS5000Admin]')
```

- Specify the remaining roles for the group, by using a separate statement for each role:

```
print AdminTask.mapGroupsToAdminRole('[-roleName klmBackup
-groupids DS5000Admin]')
print AdminTask.mapGroupsToAdminRole('[-roleName klmAudit
-groupids DS5000Admin]')
print AdminTask.mapGroupsToAdminRole('[-roleName klmView
-groupids DS5000Admin]')
print AdminTask.mapGroupsToAdminRole('[-roleName klmCreate
-groupids DS5000Admin]')
print AdminTask.mapGroupsToAdminRole('[-roleName klmModify
-groupids DS5000Admin]')
print AdminTask.mapGroupsToAdminRole('[-roleName klmDelete
-groupids DS5000Admin]')
print AdminTask.mapGroupsToAdminRole('[-roleName klmGet
-groupids DS5000Admin]')
print AdminTask.mapGroupsToAdminRole('[-roleName DS5000
-groupids DS5000Admin]')
```

where:

- **authorizationGroupName**

The name of the authorization group. If you do not specify this parameter, the cell level authorization group is assumed. (String, optional)

- **roleName**

The name of the administrative role. (String, required)

- **groupids**

The list of group IDs that are mapped to the administrative role. (String[])

3. Save your work.

- Graphical user interface:

Confirm completion of your task, by using the prompt that the graphical user interface provides.

- Command-line interface:

Save your configuration. For example, by using Jython, type:

```
print AdminConfig.save()
```

4. Ensure that the roles that you saved to the group were assigned.

- Graphical user interface

Exit and reenter the Administrative group roles page. The additional roles appear.

- Command-line interface

Using Jython syntax, type:

```
print AdminTask.listGroupIDsOfAuthorizationGroup()
```

What to do next

Next, specify other groups that your organization might require. For example, specify an administrative group to do operator tasks.

Creating a user in a group

Create a user and assign membership for the user to a group of system administrators.

About this task

This task uses the WASAdmin user ID on the WebSphere Integrated Solutions Console to create a user and add the user to a group.

Note: To access IBM Security Key Lifecycle Manager graphical user interface or command-line interface, the user must be assigned to this group:
klmGUICLIAccessGroup

For more information about the commands that create groups and users, see the IBM WebSphere Application Server documentation (http://www-01.ibm.com/support/knowledgecenter/SSAW57_8.5.5/com.ibm.websphere.nd.iseries.doc/ae/rxml_atwimmgt.html).

Procedure

1. Log on to the WebSphere Integrated Solutions Console.
 - Graphical user interface:
 - a. On the browser Welcome page, type a user ID of WASAdmin and a password value such as wasadminpw.
 - b. In the navigation tree, click **Users and Groups > Manage Users**.
 - Command-line interface:

In the *WAS_HOME/bin* directory, start a **wsadmin** session by using Jython. Log on to **wsadmin** with an authorized user ID, such as the WASAdmin user ID. For example, on Windows systems, navigate to the *drive:\Program Files (x86)\IBM\WebSphere\AppServer\bin* directory and type:

 - Windows systems:

```
wsadmin -username WASAdmin -password wasadminpw -lang jython
```
 - Systems such as AIX or Linux:

```
./wsadmin.sh -username WASAdmin -password wasadminpw -lang jython
```
2. Create a user, specifying membership in the new group.
 - Graphical user interface:
 - a. On the Manage Users page, click **Create**.
 - b. On the Create a User page, specify required information such as the user ID and password. For example, type myAdmin as a user ID, and mypwd as the password.
 - c. Click **Create**.
 - d. Click the link to the new user ID to display the user properties.
 - e. On the User Properties dialog, click **Groups**.
 - f. Click **Add**.
 - g. On the Add a User to Groups dialog, click **Search**.
 - h. In the table of groups, select the group that you previously created and click **Add**.

- i. Read the confirmation message that the user was added to the group and click **Close**.
- Command-line interface:
 - a. First, create the user. Type `createUser` and specify the required values to create a user. For example, by using Jython, type:


```
print AdminTask.createUser ('[-uid myAdmin -password tempPass
                              -confirmPassword tempPass -cn myAdmin -sn JDoe]')
```

 where:
 - uid** Specifies the unique ID for the user that you want to create. (String, required)
 - password** Specifies the password for the user. (String, required)
 - confirmPassword** Specifies the password again to validate how it was entered for the password parameter. (String, optional)
 - cn** Specifies the first name or given name of the user. (String, optional)
 - sn** Specifies the last name or family name of the user. (String, optional)
 - b. Add the user as a member of the group. For example, in Jython type:


```
print AdminTask.addMemberToGroup('[-memberUniqueName
                                   uid=myAdmin,o=defaultWIMFileBasedRealm
                                   -groupUniqueName cn=DS5000Admin,o=defaultWIMFileBasedRealm]')
```

 where:
 - memberUniqueName** *uniqueName* Specifies the unique name value for the user or group that you want to add to the specified group.
 - groupUniqueName** *uniqueName* Specifies the unique name value for the group to which you want to add the user.
3. Verify that the user is a member of the group.
 - Graphical user interface:
 - a. In the navigation tree, click **Users and Groups > Manage Users**.
 - b. On the Manage Users page, in the **User ID** column, click the entry for the new user ID.
 - c. On the User Properties dialog, click the **Groups** tab. Verify that the user is a member of the new group.
 - Command-line interface:
 For example, by using Jython, type:


```
print AdminTask.getMembersOfGroup('[-uniqueName
                                   cn=DS5000Admin,o=defaultWIMFileBasedRealm]')
```
4. Save your work.
 - Graphical user interface:
 Confirm completion of your task, by using the prompt that the graphical user interface provides.
 - Command-line interface:
 Save your configuration. For example, by using Jython, type:

```
print AdminConfig.save()
```

5. If you used the command-line interface to create the user, run the **stopServer** and **startServer** commands to restart the IBM Security Key Lifecycle Manager server. Then, log in as the new user.

What to do next

Next, validate that the user can do authorized tasks. Log out as WASAdmin. Log in as the new user and confirm that you can do tasks by using IBM Security Key Lifecycle Manager.

Validating user tasks

Validate that a new user in an administrative group can carry out tasks.

About this task

This task validates that a user in a group can do tasks that group membership provides. For example, the user can administer DS5000 storage servers.

Note: To access IBM Security Key Lifecycle Manager graphical user interface or command-line interface, the user must be assigned to this group:
klmGUICLIAccessGroup

For more information about the commands that map groups to roles, see the IBM WebSphere Application Server documentation (http://www-01.ibm.com/support/knowledgecenter/SSAW57_8.5.5/com.ibm.websphere.nd.doc/ae/rxml_atauthorizationgroup.html).

Procedure

Verify that the user can do a set of tasks that group membership provides.

- Graphical user interface:
 1. Log out of the WASAdmin user ID.
 2. Log in to the graphical user interface as an authorized user in the group. For example, log in as myAdmin.
 3. On the Key and Device Management table, verify that the only administrative choice is DS5000.

Alternatively, if your earlier tasks defined administrative activities for a new device group such as myDS5000, verify that the only administrative choice is myDS5000.
 4. Select the device and click **Go to > Manage keys and devices**.
 5. Alternatively, right-click the device and select **Manage keys and devices**.
 6. On the management page for DS5000, complete a task. For example, add a new key group.
- Command-line interface:
 1. Log out of **wsadmin** as wasadmin.
 2. In the *WAS_HOME/bin* directory, start a new **wsadmin** session by using Jython. Then, log on to **wsadmin** with an authorized user ID, such as the new myAdmin user ID. For example, on Windows systems, navigate to the directory *drive:\Program Files (x86)\IBM\WebSphere\AppServer\bin* and type:
 - Windows systems:
wsadmin -username myAdmin -password password -lang jython

- Systems such as AIX or Linux:
./wsadmin.sh -username myAdmin -password password -lang jython

3. Add an example key group. For example, type:

```
print AdminTask.tklmGroupCreate
('[-name GROUP-DS5000-abcd2de9 -type keygroup -usage DS5000]')
```

Alternatively, send the following HTTP request by using a REST client:

```
POST https://localhost:9080/SKLM/rest/v1/keygroups/newGroup
Content-Type: application/json
Accept : application/json
Authorization: SKLMAuth authId=139aeh34567m
{"usage":"DS500"}
```

What to do next

Next, specify other groups that your organization might require. For example, specify a group to do operator or auditor tasks.

Password policy for IBM Security Key Lifecycle Manager user

The password policy that applies to the password of a new IBM Security Key Lifecycle Manager user is specified by the `SKLM_HOME/config/TKLMPasswordPolicy.xml` file.

The policy does not apply to the initial passwords that are created for default users such as SKLMAdmin. These default users are created during IBM Security Key Lifecycle Manager installation.

The password policy does apply to changes to passwords for default users, and to new and changed passwords for new users. Policy checking is done only when you create or change a user profile. You must assign a role to a new user before that user attempts to log in to IBM Security Key Lifecycle Manager.

The password policy is enabled by default. You can use an XML or ASCII editor to change this file. To disable the policy, change the value of the **enabled** parameter in the policy file to false:

```
PasswordPolicy enabled="true"
```

IBM Security Key Lifecycle Manager supports these password rules:

Table 1. Password rules

Rule	Default value
Minimum length	6
Maximum length	20
Minimum number of numeric characters	2
Minimum number of alphabetic characters	3
Maximum number of consecutive occurrences of the same character	2
Disallow the presence of the user ID* in the password	Enabled
Disallow the presence of the user name* in the password	Enabled

Table 1. Password rules (continued)

Rule	Default value
<p>* Detection of this value is case-sensitive.</p> <p>Note: To specify that the value is not case-sensitive, edit the default password policy and specify <code>CaseInsensitive</code> for the user ID and user name:</p> <pre><?xml version="1.0" encoding="UTF-8"?> <PasswordPolicy version="1.0" uuid="" name="Password policy for TKLM" enabled="true"> <Description/> <PasswordRules><![CDATA[<?xml version="1.0" encoding="UTF-8"?> <PasswordRuleSet version="1.0"> <MinLengthConstraint Min="6"/> <MaxLengthConstraint Max="20"/> <MaxSequentialChars Max="2"/> <MinAlphabeticCharacters Min="3"/> <MinDigitCharacters Min="2"/> <NotUserIDCaseInsensitive/> <NotUserNameCaseInsensitive/> </PasswordRuleSet>]]></PasswordRules> </PasswordPolicy></pre>	

Changing the password policy

Use an editor to manually change the password policy that IBM Security Key Lifecycle Manager provides.

About this task

Ensure that you change only the element and attribute values in the password policy, not the element and attribute names themselves. The password policy applies to changes to passwords for default users, and to new and changed passwords for new users. Policy checking is done only when you create or change a user profile.

Procedure

1. Before you begin, make a backup copy of the `SKLM_HOME/config/TKLMPasswordPolicy.xml` file in a secure location. If a changed password policy has problems, you can revert to the backup copy.
2. Edit the `TKLMPasswordPolicy.xml` file in a text editor, changing only values of the XML elements and attributes in the password policy.
3. Save the changed file.

The policy change occurs immediately. You do not need to restart the IBM Security Key Lifecycle Manager server.

4. To test the changes, log in to WebSphere Application Server as WASAdmin and create a user profile for a new user.

Confirm that a password that meets the policy is accepted, and that a password that violates the policy is rejected. When done, if necessary, delete the test user profile.

Changing a user password

The changed password of a user must comply with the password policy that IBM Security Key Lifecycle Manager provides.

About this task

This task uses the WASAdmin user ID on the WebSphere Integrated Solutions Console to change the password of a user, including the password for the SKLMAdmin user ID.

For more information about the commands that create groups and users, see the IBM WebSphere Application Server documentation (http://www-01.ibm.com/support/knowledgecenter/SSAW57_8.5.5/com.ibm.websphere.nd.iseries.doc/ae/rxml_atwimmgt.html).

Procedure

1. Log on to the WebSphere Integrated Solutions Console.

- Graphical user interface:
 - a. On the browser Welcome page, type a user ID of WASAdmin and a password value, such as wasadminpw.
 - b. In the navigation tree, click **Users and Groups > Manage Users**.
- Command-line interface:

In the *WAS_HOME/bin* directory, start a **wsadmin** session by using Jython. Log on to **wsadmin** with an authorized user ID, such as the WASAdmin user ID. For example, on Windows systems, navigate to the *drive:\Program Files (x86)\IBM\WebSphere\AppServer\bin* directory and type:

 - Windows systems:

```
wsadmin -username WASAdmin -password wasadminpw -lang jython
```
 - Systems such as AIX or Linux:

```
./wsadmin.sh -username WASAdmin -password wasadminpw -lang jython
```

2. Change the password for a user.

- Graphical user interface:
 - a. On the **Manage Users > Search for Users** dialog, click **Search**.
 - b. In the search criteria table, double-click a selected user ID. For example, double-click myAdmin as a user ID.
 - c. On the User Properties dialog, change the value of the **Password** and **Confirm password** fields.
 - d. Click **OK**.
- Command-line interface:
 - a. Type updateUser and specify the required values. For example, by using Jython, type on one line:

```
print AdminTask.updateUser('-uniqueName uid=test2,
o=defaultWIMFileBasedRealm -password secret12 -confirmPassword secret12')
```

Where,
 - uniqueName**
Specifies the unique name for the user with a password that you want to create. (String, required)

You might use the **searchUsers** command to verify that the name correctly identifies the user before you change the password.
 - password**
Specifies the password for the user. (String, required)

The new password must comply with the password policy that IBM Security Key Lifecycle Manager provides.

-confirmPassword

Specifies the password again to validate how it was entered for the password parameter. (String, optional)

What to do next

Next, validate that the user can log in. Log out as WASAdmin. Log in as the user and confirm that the changed password is accepted.

Changing IBM Security Key Lifecycle Manager user password

You can use the IBM Security Key Lifecycle Manager application user ID to change the user password. The changed password must comply with the password policy that IBM Security Key Lifecycle Manager provides.

About this task

For more information about the commands to change passwords, see the IBM WebSphere Application Server documentation (http://www-01.ibm.com/support/knowledgecenter/SSAW57_8.5.5/com.ibm.websphere.nd.iseries.doc/ae/rxml_atwimmgt.html).

Procedure

1. Navigate to the appropriate page or directory:
 - Command-line interface:
 - In the *WAS_HOME/bin* directory, start a wsadmin session by using Jython. Log on to wsadmin with an authorized user ID.

Windows

Navigate to the C: \Program Files (x86)\IBM\WebSphere\AppServer\bin directory and type:

```
wsadmin.bat -username <SKLM user> -password <SKLM user passwd> -lang jython
```

AIX or Linux

Navigate to the /opt/IBM/WebSphere/AppServer/bin directory and type:

```
./wsadmin.sh -username <SKLM user> -password <SKLM user passwd> -lang jython
```

- Graphical user interface:
 - Log on to the graphical user interface.
2. Change the password for a user.
 - Command-line interface:
 - Run the following command:

Example:

```
AdminTask.changeMyPassword('[-oldPassword skladmin -newPassword  
Ibm12one  
-confirmNewPassword Ibm12one]')
```

- Graphical user interface:
 - a. On the header bar, click the **<SKLM User>** link.
 - b. Click **Change Password**.

- c. In the Change Password dialog, type your **Current password**.
- d. Type your **New password**.
- e. Enter the new password again in the **Confirm new password** field.
- f. Click **Change Password**.

Creating a device group

Depending on your organization requirements, you can create a device group to manage a subset of devices that have a restricted business use, such as LTO tape drives used by a single division. You must also create a role with a name that matches the name of the device group, including case. Name matching is case-sensitive.

About this task

This task uses the SKLMAdmin user ID and the IBM Security Key Lifecycle Manager interface to create an extra device group.

Your user ID must have either:

- The securityOfficer role
- Permission to the administrative actions (**k1mAdminDeviceGroup**)

If you have the **k1mAdminDeviceGroup** permission, you can create, view, and delete a device group. It is not required that you first define a role for the device group. However, your other actions are limited by the permissions that you have. For example, if you have only **k1mAdminDeviceGroup** permission, you cannot update the attributes after you create the device group.

Procedure

1. Log on to IBM Security Key Lifecycle Manager.
 - Graphical user interface:
On the browser Welcome page, type a user ID of SKLMAdmin and a password value, such as mypassword.
 - Command-line interface:
In the *WAS_HOME/bin* directory, start a **wsadmin** session by using Jython. Log on to **wsadmin** with an authorized user ID, such as the SKLMAdmin user ID. For example, on Windows systems, navigate to the *drive:\Program Files (x86)\IBM\WebSphere\AppServer\bin* directory and type:
 - Windows systems:
`wsadmin -username SKLMAdmin -password mypwd -lang jython`
 - Systems such as AIX or Linux:
`./wsadmin.sh -username SKLMAdmin -password mypwd -lang jython`
 - REST interface:
 - Open a REST client.
2. Navigate to the appropriate page or directory:
 - Graphical user interface:
Click **Advanced Configuration > Device Group**.
 - a. In the Device Group table, click **Create**.
 - b. In the Create Device Group dialog, complete the required fields and click **Create**.
 - Command-line interface, type:
`AdminTask.tklmDeviceGroupCreate('[-name myLTO -deviceFamily LTO]')`

- REST interface:
 - a. Obtain a unique user authentication identifier to access IBM Security Key Lifecycle Manager REST services. For more information about the authentication process, see Authentication process for REST services.
 - b. To invoke **Device Group Create REST Service**, send the HTTP POST request. Pass the user authentication identifier that you obtained in Step a along with the request message as shown in the following example.

```
POST https://localhost:9080/SKLM/rest/v1/deviceGroups/newGroup
Content-Type: application/json
Accept : application/json
Authorization: SKLMAuth authId=139aeh34567m
{"deviceFamily":"LT0","shortName":"myLT0","longName":"my companyname
LT0 devices"}
```

3. Verify that the device group exists.

- Graphical user interface:
On the device group management page, scan the Device Group table to locate the device group.
- Command-line interface, type:

```
print AdminTask.tklmDeviceGroupList ('[-deviceFamily myLT0 -v y]')
```
- REST interface:
Send the following HTTP GET request by using a REST client:

```
GET https://localhost:9080/SKLM/rest/v1/deviceGroups
Content-Type: application/json
Accept : application/json
Authorization: SKLMAuth authId=139aeh34567m
Accept-Language : en
```

What to do next

Create a role with a name that matches the device group.

Creating a role for a new device group

When you create a new IBM Security Key Lifecycle Manager device group, also create a role for the device group. Specify the same name for both the device group and the role, including case. Name matching is case-sensitive.

About this task

You can add the role for a device group to the WebSphere Application Server by editing the `admin-authz.xml` configuration file.

Procedure

1. On Windows operating system, edit the `<WAS_HOME>/profiles/KLMProfile/cofig/cells/SKLMSKCell/admin-authz.xml` file by adding the following lines:

```
<roles xmi:id=<roleId> roleName=<deviceGroupName>/>
<authorizations xmi:id=<roleAssignmentId> role=<roleId>/>
```

The values for `roleId` and `roleAssignmentId` must be unique across the roles and authorizations that are exists in the `admin-authz.xml` file.

For example, you must add the following lines if a new device group, such as `MyDS5K` is added:

```
<roles xmi:id="MyDS5K_Role" roleName="MyDS5K"/>
<authorizations xmi:id="MyDS5K_Role_Auth" role="MyDS5K_Role"/>
```

2. Restart WebSphere Application Server. You must stop the server and then restart. For instructions about how to stop and start the server, see “Starting and stopping the IBM Security Key Lifecycle Manager server on distributed systems” on page 120.

What to do next

Next, you might specify that a user group has permissions to the new device group and the necessary administrative tasks, such as view or configure.

Database administration

The installation process provides a default Administrator user ID with the necessary permissions and password.

You must ensure that the user ID remains active and complies with the security policy that is active on the system.

Moving DB2 transaction log files for good performance

Periodically move old DB2[®] transactional logs that the IBM Security Key Lifecycle Manager database creates. Otherwise, large numbers of transactional logs might affect performance.

About this task

DB2 transactional logs occur in these directories:

Windows systems:

```
INSTANCEHOME:\sklmbarchive\SKLMB26\SKLMB26\NODE0000\LOGSTREAM0000\C0000000
```

where:

- *INSTANCEHOME* is the drive letter that you specified during the installation.
- SKLMB26 is the database instance owner.
- SKLMB26 is the name of the IBM Security Key Lifecycle Manager database.
- NODE0000, LOGSTREAM0000, and C0000000 might be different on your system.

Systems such as Linux or AIX:

```
~sklmbarchive/sklmb26/SKLMB26/NODE0000/LOGSTREAM0000/C0000000
```

where:

- sklmb26 is the database instance owner.
- SKLMB26 is the name of the IBM Security Key Lifecycle Manager database.
- NODE0000, LOGSTREAM0000, and C0000000 might be different on your system.

If IBM Security Key Lifecycle Manager manages many keys and if the disk partition that contains the sklmbarchive directory has low free disk space, move the old transaction logs to a different disk partition.

Note: As you carry out this task, be careful not to move the current active log.

Take these steps on a periodic basis:

Procedure

1. Create an IBM Security Key Lifecycle Manager backup by using the graphical user interface, command-line interface, or REST interface. Otherwise, the next backup might fail.
2. Log in as the database instance owner on systems such as Linux or AIX, or the DB2 administrator on Windows systems.
3. Create a directory on another partition that has adequate disk space to which you can move old log files.
4. Identify the first active log. Type:

Windows systems:

```
db2cmd
SET DB2INSTANCE=sk1mdb26
db2 get db cfg for SKLMB26
```

Systems such as Linux or AIX:

```
db2 get db cfg for SKLMB26
```

The value for the configuration parameter First active log file identifies the first active log.

5. Move the log files that are modified earlier than the first active log from the sk1mdbarchive directory to the new directory.

Logs are named *Snnnnnnn.LOG*. Usually, the lower numbered logs are created earlier than higher numbered logs. The exception is if the database already created a log named *S99999999.LOG*. In this case, the numbering restarts at *S00000000.LOG*.

Note: Running a restore operation removes the sk1mdbarchive directory and creates a new directory.

DB2 password security issues on Windows systems

On Windows systems, the DB2 Administrator user ID and password are subject to the security policy that is active on the system.

If there is a password expiration restriction in effect, you must change the login password and DB2 password for the Administrator user ID before the expiration period expires.

In addition, the login password for the DB2 Administrator user ID and the DB2 data source password that is used by WebSphere Application Server must be the same. When you change one, you must change the other.

To change the DB2 database password, take these steps:

1. Stop the WebSphere Application Server and *all* Windows services that are related to DB2.
2. Open the Windows user management tool by opening the Control Panel and clicking **Administrative tools > Computer Management > Local Users and Groups > Users**.
3. Change the password for the IBM Security Key Lifecycle Manager database owner.
4. Open the Windows Services console by opening the Control Panel and clicking **Administrative Tools > Computer Management**.
5. On the following services, change the password by using the **Logon** tab of the **Properties** dialog box:

- DB2 - DBSKLMV26 - *sklminstance*

For example, the value of *sklminstance* might be:

```
DB2 - DBSKLMV26 - DBSKLM26
DB2 - DBSKLMV26 - SKLMD26
```

For example, with the default instance name, the value of *sklminstance* is:

```
DB2 - DBSKLMV26 - SKLMD26
```

- DB2 Governor (DBSKLMV26)
- DB Remote Command Server (DBSKLMV26)
- DB2DAS - DB2DAS00

When the passwords are changed for all the services, restart the services.

The following services must be stopped and restarted. Password change is not required:

- DB2 License Server (DBSKLMV26)
- DB2 Management Service (DBSKLMV26)

6. Start the WebSphere Application Server.
7. Using the **wsadmin** interface that the WebSphere Application Server provides, specify the Jython syntax.

```
wsadmin -username WASAdmin -password mypwd -lang jython
```
8. Use the **wsadmin** command to change the password of the WebSphere Application Server data source:

- a. The following command lists JAASAuthData entries:

```
wsadmin>print AdminConfig.list('JAASAuthData')
```

The result might be:

```
(cells/SKLMCell|security.xml#JAASAuthData_1379859888963)
```

- b. Identify the data source ID with the alias that matches the string *sklm_db*. Also, identify the data source ID with the alias that matches the string *sklmdb*:

```
print AdminConfig.showAttribute('JAASAuthData_list_entry', 'alias')
```

For example, type on one line:

```
print AdminConfig.showAttribute
('(cells/SKLMCell|security.xml#JAASAuthData_1379859888963)', 'alias')
```

The result is:

```
sklm_db
```

- c. Change the password of the *sklm_db* alias, entering this command on one line:

```
print AdminConfig.modify('JAASAuthData_list_entry',
'[[password newpassword]]')
```

If you specify special characters in the password, use quotation marks as delimiters when you specify the password value.

For example, type on one line:

```
print AdminConfig.modify
('(cells/SKLMCell|security.xml#JAASAuthData_1379859888963)',
'[[password tucs0naz]]')
```

- d. Save the changes:

```
print AdminConfig.save()
```

- e. Stop and restart the IBM Security Key Lifecycle Manager server by using the **stopServer** and **startServer** commands.

Alternatively, stop and restart the IBM Security Key Lifecycle Manager server by using Windows Computer Management.

- 1) Open the Control Panel and click **Administrative Tools > Computer Management > Services and Applications > Services**.
 - 2) Stop and start the IBM Security Key Lifecycle Manager server service, which has a name like IBM WebSphere Application Server V8.5 - SKLM26Server.
- f. Verify that you can connect to the database by using the WebSphere Application Server data source.
- 1) First, type:


```
print AdminConfig.list('DataSource')
```

 The result might be:


```
"Default Datasource(cells/SKLMCell/nodes/SKLMNode/servers/server1|resources.xml#DataSource_1183122153625)"
"SKLM DataSource(cells/SKLMCell/nodes/SKLMNode/servers/server1|resources.xml#DataSource_1379859893896)"
"SKLM scheduler XA Datasource(cells/SKLMCell/nodes/SKLMNode/servers/server1|resources.xml#DataSource_1379859896273)"
DefaultEJBTimerDataSource(cells/SKLMCell/nodes/SKLMNode/servers/server1|resources.xml#DataSource_1000001"
```
 - 2) Test the connection on the first data source. For example, type:


```
print AdminControl.testConnection('SKLM DataSource(cells....)')
```

 For example, type on one line:


```
print AdminControl.testConnection
(' (SKLM DataSource(cells/SKLMCell/nodes/SKLMNode/servers/server1|resources.xml#DataSource_1379859893896) ')
```
 - 3) Test the connection on the remaining data source. For example, type:


```
print AdminControl.testConnection
(' (SKLM scheduler XA Datasource(cells/SKLMCell/nodes/SKLMNode/servers/server1|resources.xml#DataSource_1379859896273) ')
```
 - 4) In both cases, you receive a message that the connection to the data source was successful. For example:


```
WASX7217I: Connection to provided datasource was successful.
```

 Now you can run an IBM Security Key Lifecycle Manager operation.

DB2 password security issues on systems such as Linux or AIX

On systems such as Linux or AIX, you might want to change the password for the DB2 Administrator user ID. The login password for the DB2 Administrator user ID and the DB2 password for the user ID must be the same.

The IBM Security Key Lifecycle Manager installation program installs DB2 and prompts the installing person for a password for the user named sklmbd26. Additionally, the DB2 application creates an operating system user entry named sklmbd26. For example, the password for this user might expire, requiring you to resynchronize the password for both user IDs.

Before you can change the password of the DB2 Administrator user ID, you must change the password for the system user entry. Take these steps:

1. Log on to IBM Security Key Lifecycle Manager server as root.
2. Change user to the sklmbd26 system user entry. Type:


```
su sklmbd26
```
3. Change the password. Type:


```
passwd
```

Specify the new password.

4. Exit back to root.

```
exit
```

5. In the *WAS_HOME/bin* directory, use the **wsadmin** interface that the WebSphere Application Server provides to specify the Jython syntax.

```
./wsadmin.sh -username WASAdmin  
-password mypwd -lang jython
```

6. Change the password for the WebSphere Application Server data source:

- a. The following command lists the JAASAuthData entries:

```
wsadmin>print AdminConfig.list('JAASAuthData')
```

The result might like this example:

```
(cells/SKLMCell|security.xml#JAASAuthData_1228871756187)  
(cells/SKLMCell|security.xml#JAASAuthData_1228871757843)
```

- b. Type the **AdminConfig.showall** command for each entry to locate the alias *sklm_db*. For example, type on one line:

```
print AdminConfig.showall  
( '(cells/SKLMCell|security.xml#JAASAuthData_1228871756187)' )
```

The result is like this example:

```
{alias sklm_db}  
{description "SKLM database user j2c authentication alias"}  
{password *****}  
{userId sklmb26}
```

And also type on one line:

```
print AdminConfig.showall  
( '(cells/SKLMCell|security.xml#JAASAuthData_1228871757843)' )
```

The result is like this example:

```
{alias sklmb26}  
{description "SKLM database user J2C authentication alias"}  
{password *****}  
{userId sklmb26}
```

- c. Change the password for the *sklm_db* alias that has the identifier **JAASAuthData_1228871756187**:

```
print AdminConfig.modify('JAASAuthData_list_entry', '[[password  
passw0rdc]]')
```

For example, type on one line:

```
print AdminConfig.modify  
( '(cells/SKLMCell|security.xml#JAASAuthData_1228871756187)',  
'[[password tucs0naz]]' )
```

- d. Change the password for the *sklmb26* alias that has the identifier **JAASAuthData_1228871757843**:

```
print AdminConfig.modify('JAASAuthData_list_entry', '[[password  
passw0rdc]]')
```

For example, type on one line:

```
print AdminConfig.modify  
( '(cells/SKLMCell|security.xml#JAASAuthData_1228871757843)',  
'[[password tucs0naz]]' )
```

- e. Save the changes:

```
print AdminConfig.save()
```

- f. Exit back to root.

```
exit
```

- g. In the *WAS_HOME/bin* directory, stop the WebSphere Application Server application. For example, as WASAdmin, type on one line:

```
stopServer.sh server1 -username wasadmin -password passw0rd
```

The result is like this example:

```
ADMU0116I: Tool information is being logged in file
//opt/IBM/WebSphere/AppServer/profiles/KLMPProfile/logs/server1/stopServer.log
ADMU0128I: Starting tool with the WASProfile profile
ADMU3100I: Reading configuration for server: server1
ADMU3201I: Server stop request issued. Waiting for stop status.
ADMU4000I: Server server1 stop completed.
```

- h. Start the WebSphere Application Server application. As the WebSphere Application Server administrator, type on one line:

```
startServer.sh server1
```

- i. In the `WAS_HOME/bin` directory, use the **wsadmin** interface that the WebSphere Application Server provides to specify the Jython syntax.

```
./wsadmin.sh -username wasadmin -password mypwd -lang jython
```

- j. Verify that you can connect to the database by using the WebSphere Application Server data source.

- 1) First, query for a list of data sources. Type:

```
print AdminConfig.list('DataSource')
```

The result might be like this example:

```
"SKLM DataSource(cells/SKLMCell/nodes/SKLMNode/
servers/server1|resources.xml#DataSource_1228871762031)"
"SKLM scheduler XA Datasource(cells/SKLMCell/nodes/SKLMNode/
servers/server1|resources.xml#DataSource_1228871766562)"
"Tivoli Common Reporting Data Source(cells/SKLMCell|resources.xml#
DataSource_1227211230078)"
DefaultEJBTimerDataSource(cells/SKLMCell/nodes/SKLMNode/
servers/server1|resources.xml#DataSource_1000001)
ttssdb(cells/SKLMCell|resources.xml#DataSource_1227211144390)
```

- 2) Type:

```
print AdminControl.testConnection('SKLM DataSource(cells....)')
```

For example, type on one line:

```
print AdminControl.testConnection
('SKLMDataSource(cells/SKLMCell/nodes/SKLMNode/
servers/server1|resources.xml#DataSource_1228871762031)')
```

- 3) Test the connection on the remaining data source. For example, type:

```
print AdminControl.testConnection
('SKLM scheduler XA Datasource(cells/SKLMCell/nodes/SKLMNode/
servers/server1|resources.xml#DataSource_1228871766562)')
```

- 4) In both cases, you receive a message that the connection to the data source was successful. For example:

```
WASX7217I: Connection to provided data source was successful.
```

Stopping the DB2 server

To stop the database server, stop the WebSphere Application Server and stop the DB2 server.

About this task

You must be the database instance owner on systems such as AIX or Linux, or the Local Administrator on Windows systems.

To stop the database server, take these steps:

Procedure

1. Log in as the database instance owner on systems such as AIX or Linux, or log in as Local Administrator on Windows systems.
2. Stop the WebSphere Application Server. Type this command:

Windows systems:

```
cd C:\Program Files (x86)\IBM\WebSphere\AppServer\bin
.\stopServer.bat server1 -username wasadmin -password mysecretpwd
```

Systems such as AIX or Linux:

```
/opt/IBM/WebSphere/AppServer/bin/stopServer.sh server1 -username wasadmin
-password mysecretpwd
```

3. Stop the DB2 server. Type these commands:

Windows systems:

```
set DB2INSTANCE=sk1mdb2
db2stop
```

Systems such as AIX or Linux:

```
su -sk1mdb2
db2stop
```

Changing the DB2 server host name

After you change the IBM Security Key Lifecycle Manager system host name, you might want to change the host name of the DB2 server.

About this task

Obtain the current steps to change the host name for your level of the DB2 server from the technote at this web address: http://www.ibm.com/support/docview.wss?rs=71&context=SSEPGG&context=SSEPDU&context=SSVGXH&context=SSVGZB&context=SSFHEG&context=SSYK8P&context=SSTLZ9&q1=db2+change+hostname&uid=swg21258834&loc=en_US&cs=utf-8&lang=en

Changing an existing WebSphere Application Server host name

You must change the host name of WebSphere Application Server before you change the system host name.

Procedure

1. Change the host name of WebSphere Application Server. For more information about how to change the host name, see IBM WebSphere Application Server documentation (http://www-01.ibm.com/support/knowledgecenter/SSEQTP_8.5.5/com.ibm.websphere.base.iseries.doc/ae/tagt_hostname.html).
2. When this task succeeds, change the host name of the DB2 server. For more information, see “Changing the DB2 server host name.”

Accepting pending devices

Use the device pending function to accept or reject a device that contacts IBM Security Key Lifecycle Manager.

About this task

You can use the Pending Device Requests page or you can use several commands to accept or reject a device that contacts IBM Security Key Lifecycle Manager. If the

device belongs to the DS5000 device family, and machine affinity is enabled, you might also accept or reject a relationship between a device and a machine. Using machine affinity, you can restrict key serving to specific device and machine combinations.

Procedure

1. Keys are auto-generated for a device in a DS5000 device group when a pending request arrives. Carry out a backup before you accept the device to ensure that keys are backed up before served to a device. For more information, see the administering backup and restore files.

2. Navigate to the appropriate page or directory:

- Graphical user interface:

Log on to the graphical user interface. From the navigation tree, click **IBM Security Key Lifecycle Manager**.

- Command-line interface:

In the *WAS_HOME/bin* directory, start a **wsadmin** session by using Jython. Log on to **wsadmin** with an authorized user ID, such as the SKLMAdmin user ID. For example, on Windows systems, navigate to the *drive:\Program Files (x86)\IBM\WebSphere\AppServer\bin* directory and type:

- Windows systems:

```
wsadmin -username SKLMAdmin -password mypwd -lang jython
```

- Systems such as AIX or Linux:

```
./wsadmin.sh -username SKLMAdmin -password mypwd -lang jython
```

3. If you are not previously determined how to accept pending devices, set the **device.AutoPendingAutoDiscovery** attribute to a value that adds incoming devices to the pending devices list.

Specify a setting such as 2 (auto pending). All incoming devices are added to a pending list, but are not automatically served keys upon request. You must accept or reject a device in the pending devices list before the device is served keys upon request. Do not use a setting of 1 (auto accept) for the DS5000 device family. This setting allows generation and serving of keys to DS5000 storage servers before you backup data.

- Graphical user interface:

- a. Navigate to the Key and Device Management page for the device group of the pending devices.
- b. In the drop-down list at the bottom of the page, select **Hold new device requests pending my approval**.

- Command-line interface:

For example, for a DS5000 device, type:

```
print AdminTask.tklmDeviceGroupAttributeUpdate ('[-name DS5000  
-attributes "{device.AutoPendingAutoDiscovery 2}"']')
```

4. List the pending devices.

- Graphical user interface:

Navigate to the Welcome page. In the Action Items area, click the pending devices link.

- Command-line interface:

Type:

```
print AdminTask.tklmPendingDeviceList ('[-usage DS5000]')
```

5. Approve or reject a pending device request.

- Graphical user interface:

In the Pending Device Requests table, select a pending device and click **Accept** or **Reject**.

A pending request is listed only once for a DS5000 device that also has a machine-device relationship. The request appears with the table with a value for the machine ID. Accepting the pending device request also accepts the machine-device relationship.

On the Accept Device Request dialog, click **Accept** or **Modify and Accept**. If you choose to modify the pending device information, make the necessary changes and click **Accept**.

- Command-line interface:

- You might use one command to accept a pending DS5000 device and also the pending machine-device relationship. For example, type:

```
print AdminTask.tklmPendingMachineDeviceAccept
('[-deviceUUID DEVICE-7d588437-e725-48bf-a836-00a47df64e78
-machineID 304238303030343700000000000000]')
```

- Otherwise, you might first accept a pending device, assigning the device to the appropriate device group. To accept a pending DS5000 device and later accept a machine-device relationship, for example

- a. First, type:

```
print AdminTask.tklmPendingDeviceAccept
('[-deviceUUID DEVICE-7d588437-e725-48bf-a836-00a47df64e78
-usage DS5000]')
```

- b. Later, accept or reject pending relationships between a device and a machine.

- 1) List all of the pending devices that have a relationship with a machine ID, or all devices, if no machine ID is specified. For example, type:

```
print AdminTask.tklmPendingMachineDeviceList
('[-machineID 304238303030343700000000000000]')
```

- 2) Accept or reject a pending device and machine relationship. Acceptance writes the relationship data to the IBM Security Key Lifecycle Manager data store.

```
print AdminTask.tklmPendingMachineDeviceAccept
('[-deviceUUID DEVICE-7d588437-e725-48bf-a836-00a47df64e78
-machineID 304238303030343700000000000000]')
```

- 6. A success indicator varies, depending on the interface:

- Graphical user interface:

The accepted device is listed in the appropriate management page for the device group. For example, the drive serial number of an LTO tape drive is visible in a column of the table on the management page.

- Command-line interface:

A completion message indicates success.

What to do next

Examine the list of accepted devices. Use these commands:

- **tklmDeviceList** to list information about all devices of the specified device type.
- **tklmMachineDeviceList** to list all the devices that are associated with a specific machine ID, or all devices, if no machine ID is specified.

Moving devices between device groups

Use the device update function to move device from one existing device group to another existing device group. For example, you might want to move a device to the MYDS5000 device group.

About this task

You can use the Modify Device page, **tklmDeviceUpdate** command, or **Device Update REST Service** to move a device that contacts IBM Security Key Lifecycle Manager from one device group to another within the same device family. For example, you might want to move a device to the MYDS5000 device group within the DS5000 device family.

For more information about creating a device group, see “Creating a device group” on page 33.

Procedure

1. Navigate to the appropriate page or directory:
 - Graphical user interface:
 - a. Log on to the graphical user interface.
 - b. In the Key and Device Management section on Welcome page, select **DS5000**.
 - c. Right-click **DS5000**.
 - d. Click **Manage keys and devices**.
 - Command-line interface:

In the *WAS_HOME/bin* directory, start a **wsadmin** session by using Jython. Log on to **wsadmin** with an authorized user ID, such as the SKLMAdmin user ID. For example, on Windows systems, navigate to the *drive:\Program Files (x86)\IBM\WebSphere\AppServer\bin* directory and type:

 - Windows systems:

```
wsadmin -username SKLMAdmin -password mypwd -lang jython
```
 - Systems such as AIX or Linux:

```
./wsadmin.sh -username SKLMAdmin -password mypwd -lang jython
```
 - REST interface:
 - Open a REST client.
2. Locate the device that you want to move to another device group within a parent device family.
 - Graphical user interface:

On the Key and Device Management DS5000 page, locate the device in the device table. For example, the device might have a serial number such as aaa123.
 - Command-line interface:

Type the following command:

```
print AdminTask.tklmDeviceList ('[-type DS5000]')
```

In the command output, locate the value of the device uuid. For example:

```
Description = My long description
Serial Number = aaa123
Device uuid = DEVICE-b7678b4d-3898-4f8c-9557-dbb2f381fc8a
```



```
Device group = DS5000
Device Text =
World wide name =
Sym alias = DS5K-aaa123
```

- REST interface:
 - a. Obtain a unique user authentication identifier to access IBM Security Key Lifecycle Manager REST services. For more information about the authentication process, see Authentication process for REST services.
 - b. To invoke **Device List Type REST Service**, send the HTTP GET request. Pass the user authentication identifier that you obtained in Step a along with the request message as shown in the following example.

```
GET https://localhost:9080/SKLM/rest/v1/devices?type=DS5000
Content-Type: application/json
Accept : application/json
Authorization : SKLMAuth userAuthId=37ea1939-1374-4db7-84cd-14e399be2d20
Accept-Language : en
```

In the success response locate the value of the device uuid. For example:

```
Status Code : 200 OK
Content-Language: en
[
{
  "Description": "My long description",
  "Serial Number": "aaa123",
  "Device uuid": "DEVICE-b7678b4d-3898-4f8c-9557-dbb2f381fc8a",
  "Device group": "DS5000",
  "World wide name": "",
  "Sym alias": "DS5K-aaa123"
},
]
```

3. Ensure that the target device group exists.

- Graphical user interface:

On the Key and Device Management DS5000 page, in the device table, select the device and click **Modify > Device**.

On the Modify Device page, in the **Currently assigned device group** field, expand the list to determine whether the **MYDS5000** device group is available.

- Command-line interface:

Type the following command:

```
print AdminTask.tklmDeviceGroupList ('[-deviceFamily DS5000 -v y]')
```

Locate the device group. For example:

Device Group UUID	10000
Device Group Name	MYDS5000
Device Family	DS5000
symmetricKeySet	null
drive.default.alias1	null
drive.default.alias2	null
shortName	MYDS5000group
longName	my companyname DS5000 devices
roleName	MYDS5000
device.AutoPendingAutoDiscovery	0
enableKMIPDelete	false

- REST interface:

Send the following HTTP request:

```
GET https://localhost:9080/SKLM/rest/v1/deviceGroups?deviceFamily=DS5000
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth authId=139aeh34567m
Accept-Language : en
```

Locate the device group. For example:

```
Status Code : 200 OK
Content-Language: en
[
{
  "Device Group UUID": "10000",
  "Device Group Name": "MYDS5000",
  "Device Family": "DS5000",
  "symmetricKeySet": null,
  "drive.default.alias1": null,
  "drive.default.alias2": null,
  "shortName": MYDS5000group,
  "longName": my companyname DS5000 devices,
  "roleName": "MYDS5000",
  "device.AutoPendingAutoDiscovery": "0",
  "enableKMIPDelete": "false"
},
]
```

4. Update the device to specify the new device group.

- Graphical user interface:

On the Modify Device page, in the **Currently assigned device group** field, select the **MYDS5000** device group

Click **Modify Device**.

- Command-line interface:

Type the following command:

```
print AdminTask.tklmDeviceUpdate
('[-uuid DEVICE-b7678b4d-3898-4f8c-9557-dbb2f381fc8a -type MYDS5000]')
```

- REST interface:

Send the following HTTP request:

```
PUT https://localhost:9080/SKLM/rest/v1/devices
Content-Type: application/json
Accept : application/json
Authorization: SKLMAuth authId=139aeh34567m
{"uuid":"DEVICE-b7678b4d-3898-4f8c-9557-dbb2f381fc8a","type":
"MYDS5000"}
```

5. Validate that the device is in the new device group.

- Graphical user interface:

On the Key and Device Management DS5000 page, the device is no longer listed in the device table. Open the Key and Device Management MYDS5000 page and ensure that the device is listed in the device table.

- Command-line interface:

Type the following command:

```
print AdminTask.tklmDeviceList ('[-type MYDS5000]')
```

For example, the output contains the uuid value of the device and the name of the new device group:

```
Description = My long description
Serial Number = aaal23
Device uuid = DEVICE-b7678b4d-3898-4f8c-9557-dbb2f381fc8a
```

```
Device group = MYDS5000
Device Text =
World wide name =
Sym alias = DS5K-aaa123
```

- REST interface:

Send the following HTTP request:

```
GET https://localhost:9080/SKLM/rest/v1/devices?type=MYDS5000
Content-Type: application/json
Accept : application/json
Authorization : SKLMAuth userAuthId=37ea1939-1374-4db7-84cd-14e399be2d20
Accept-Language : en
```

Success response contains the uuid value of the device and the name of the new device group as shown in the following example:

```
Status Code : 200 OK
Content-Language: en
[
{
  "Description": "My long description",
  "Serial Number": "aaa123",
  "Device uuid": "DEVICE-b7678b4d-3898-4f8c-9557-dbb2f381fc8a",
  "Device group": "MYDS5000",
  "World wide name": "",
  "Sym alias": "DS5K-aaa123"
},
]
```

LTO tape drive management

You can manage LTO tape drives by using IBM Security Key Lifecycle Manager.

Guided steps to create key groups and drives

When you first create key groups and drives, and later when you add more key groups or drives, IBM Security Key Lifecycle Manager provides a guided set of steps to complete the task.

Descriptions of some steps might mention command-line alternatives to do the same task. In a guided set of tasks, use the graphical user interface to complete the tasks.

Creating a key group

As a first activity, you might create keys and key groups for IBM Security Key Lifecycle Manager.

About this task

You can use the Create Key Group dialog. Alternatively, you can use the **tklmGroupCreate** command or **Group Create REST Service** to create a group to which you want to add keys. Then, use the **tklmSecretKeyCreate** command or **Secret Key Create REST Service** to create one or more symmetric keys in the existing group. Your role must have a permission to the create action and a permission to the appropriate device group.

Before you begin, determine the quantity of keys and the purpose of individual key groups that your organization requires.

Procedure

1. Navigate to the appropriate page or directory:
 - Graphical user interface:
 - a. Log on to the graphical user interface.
 - b. In the Key and Device Management section on Welcome page, select **LTO**.
 - c. Click **Go to > Guided key and device creation**.
 - d. Alternatively, right-click **LTO** and select **Guided key and device creation**.
 - Command-line interface:

In the *WAS_HOME/bin* directory, start a **wsadmin** session by using Jython. Log on to **wsadmin** with an authorized user ID, such as the SKLMAdmin user ID. For example, on Windows systems, navigate to the *drive:\Program Files (x86)\IBM\WebSphere\AppServer\bin* directory and type:

 - Windows systems:

```
wsadmin -username SKLMAdmin -password mypwd -lang jython
```
 - Systems such as AIX or Linux:

```
./wsadmin.sh -username SKLMAdmin -password mypwd -lang jython
```
 - REST interface:
 - Open a REST client.
2. Create a key group:
 - Graphical user interface:
 - a. On the Step 1: Create Key Groups page, click **Create** on the **Key Group** table.
 - b. On the Create Key Group dialog, specify values for the required and optional parameters. For example, you might create a key group with 100 keys.
 - c. Click **Create Key Group**.
 - Command-line interface:
 - a. First, create a group to which you might add keys.

Type **tklmGroupCreate** to create a group. For example, type:

```
print AdminTask.tklmGroupCreate  
('[-name GROUP-myKeyGroup -type keygroup -usage LTO]')
```
 - b. Next, use the **tklmGroupList** command obtain the value of the uuid for the group that you created. For example, type:

```
print AdminTask.tklmGroupList  
('[-name GROUP-myKeyGroup -type keygroup -v y]')
```
 - c. Then, create a group of keys and store them in the group. For example, type:

```
print AdminTask.tklmSecretKeyCreate ('[-alias abc  
-keyStoreName defaultKeyStore  
-numOfKeys 10 -usage LTO  
-keyGroupUuid KEYGROUP-316408ac-f433-4c11-92bc-0de46d05bee9]')
```
 - REST interface:
 - a. Obtain a unique user authentication identifier to access IBM Security Key Lifecycle Manager REST services. For more information about the authentication process, see Authentication process for REST services.
 - b. To invoke **Group Create REST Service**, send the HTTP POST request. Pass the user authentication identifier that you obtained in Step a along with the request message as shown in the following example.

```
POST https://localhost:9080/SKLM/rest/v1/keygroups/newGroup
Content-Type: application/json
Accept : application/json
Authorization: SKLMAuth authId=139aeh34567m
{"usage":"LTO"}
```

- c. Use **Group List REST Service** to obtain the value of the uuid for the group that you created. For example,

```
GET https://localhost:9080/SKLM/rest/v1/keygroups?name=newGroup
Content-Type: application/json
Accept : application/json
Authorization: SKLMAuth authId=139aeh34567m
Accept-Language : en
```

- d. Then, create a group of keys and store them in the group by using **Secret Key Create REST Service**. For example, you can send the following HTTP request:

```
POST https://localhost:9080/SKLM/rest/v1/keys
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth authId=139aeh34567m
{"alias":"abc","numOfKeys":"10","KEYGROUP-316408ac-f433-4c11-92bc-0de46d05bee9","usage":"LTO"}
```

3. A success indicator varies, depending on the interface:

- Graphical user interface:
The key group is listed as an item in the **Key Group** table.
- Command-line interface:
A completion message indicates success.
- Rest interface:
The status code 200 OK indicates success.

What to do next

Back up new keys before the keys are served to devices. You might also go to the next guided step to define specific devices, and associate key groups with the devices. Select **Step 2: Identify Drives** or click **Go to Next Step**.

Identifying drives

You might identify an LTO tape drive for use with IBM Security Key Lifecycle Manager.

About this task

You can use the Add Tape Drives dialog, the **tklmDeviceAdd** command, or **Device Add REST Service** to add a device. Your role must have a permission to the create action and a permission to the appropriate device group.

Before you begin, create the key groups that you want to associate with tape drives that you identify. Additionally, determine whether you want IBM Security Key Lifecycle Manager to automatically accept requests from all drives. For greater security, after all drives are discovered, you might turn off this option for a production environment.

Procedure

1. Navigate to the appropriate page or directory:
 - Graphical user interface:
 - a. Log on to the graphical user interface.

- b. In the Key and Device Management section on Welcome page, select **LTO**.
 - c. Click **Go to > Guided key and device creation**.
 - d. Alternatively, right-click **LTO** and select **Guided key and device creation**.
- Command-line interface:
 In the *WAS_HOME/bin* directory, start a **wsadmin** session by using Jython. Log on to **wsadmin** with an authorized user ID, such as the SKLMAdmin user ID. For example, on Windows systems, navigate to the *drive:\Program Files (x86)\IBM\WebSphere\AppServer\bin* directory and type:
 - Windows systems:
`wsadmin -username SKLMAdmin -password mypwd -lang jython`
 - Systems such as AIX or Linux:
`./wsadmin.sh -username SKLMAdmin -password mypwd -lang jython`
- REST interface:
 - Open a REST client.
2. Skip the Create Key Groups page. Click the **Go to Next Step** link or click **Step 2: Identify Drives**.
3. You might specify that IBM Security Key Lifecycle Manager holds new device requests for your approval.
 - Graphical user interface:
 Select **Hold new device requests pending my approval**.
 - Command-line interface:
 Use the **tklmDeviceGroupAttributeUpdate** command or **Device Group Attribute Update REST Service** to set the value of the **device.AutoPendingAutoDiscovery** attribute. For example, type:


```
print AdminTask.tklmDeviceGroupAttributeUpdate ('[-name LTO
          -attributes "{device.AutoPendingAutoDiscovery 2}"]')
```

 For an LTO device group, use the **tklmDeviceGroupAttributeUpdate** command to specify a key group by using the **symmetricKeySet** attribute in the IBM Security Key Lifecycle Manager database.
 - REST interface:
 - a. Obtain a unique user authentication identifier to access IBM Security Key Lifecycle Manager REST services. For more information about the authentication process, see Authentication process for REST services.
 - b. To invoke **Device Group Attribute Update REST Service** and to set the value of the **device.AutoPendingAutoDiscovery** attribute, send the HTTP PUT request. Pass the user authentication identifier that you obtained in Step a along with the request message as shown in the following example.


```
PUT https://localhost:9080/SKLM/rest/v1/deviceGroupAttributes
Content-Type: application/json
Accept : application/json
Authorization: SKLMAuth authId=139aeh34567m
{"name":"LTO","attributes":"device.AutoPendingAutoDiscovery 2"}
```
4. Add a device:
 - Graphical user interface:
 - a. On the Step 2: Identify Drives page, in the **Devices** table, click **Add**.
 - b. On the Add Tape Drive dialog, type the required and optional information.
 - c. Click **Add Tape Drive**.

- Command-line interface:

Type `tklmDeviceAdd` to add a device. You must specify the device group and serial number. For example, type:

```
print AdminTask.tklmDeviceAdd ('[-type LTO -serialNumber FAA49403AQJF
  -attributes "{worldwideName ABCdeF1234567890}
  {description salesDivisionDrive} {symAlias LTOKeyGroup1}"]')
```

- REST interface:

You can use **Device Add REST Service** to add a device. For example, you can send the following HTTP request:

```
POST https://localhost:9080/SKLM/rest/v1/devices
Content-Type: application/json
Accept : application/json
Authorization : SKLMAuth userAuthId=37ea1939-1374-4db7-84cd-14e399be2d20
Accept-Language : en
{"type":"LTO","serialNumber":"FAA49403AQJF","attributes":"worldwideName
ABCdeF1234567890,description marketingDivisionDrive"}
```

5. A success indicator varies, depending on the interface:

- Graphical user interface:

The device is added to the **Devices** table.

- Command-line interface:

A completion message indicates success.

- REST interface:

The status code 200 OK indicates success.

What to do next

Next, you might use the LTO Key and Device Management page to view all key groups and devices.

Managing keys, key groups, and drives

To administer keys, key groups, and devices, you map key groups to drives. You might add, modify, or delete specific keys, key groups, or devices.

About this task

Use the LTO Key and Device Management to map key groups to drives. You might add, modify, or delete specific keys, key groups, or devices. Your role must have a permission to the view action and a permission to the appropriate device group.

To change the view of information, select:

View Key Groups and Drives

View the key group names and drive serial numbers. Additionally, this view lists the key group, key, or system default that a drive uses.

View Keys, Key Group Membership and Drives

View the keys and key membership in key groups. Additionally, this view lists drive serial numbers and the key group, key, or system default that a drive uses.

Before you begin, examine the columns on the page, which provides buttons to add, modify, or delete a table item. To sort information, click a column header.



The table is organized in these areas:

- In left columns, information about keys or key groups.

For a key, the information indicates in which key group the key is a member.
For a key group, the information indicates whether the key group is used as the default, and the number of keys in the group.

- In right columns, information about drives.
The information indicates the drive serial number and the key group or specific key that the drive uses. For example, a drive might use the System Default key group.
- Icons indicate the type of keys.

Table 2. Icons and their meanings

Icon	Description
	A symmetric key or private key. A private key is an asymmetric key in a key pair with a public key and a private key.
	A key group

Procedure

1. Log on to the graphical user interface:
 - a. In the Key and Device Management section on Welcome page, select **LTO**.
 - b. Click **Go to > Manage keys and devices**.
 - c. Alternatively, right-click **LTO** and select **Manage keys and devices**.


Descriptions of some steps describe alternatives by using the graphical user interface, command-line interface, or the REST interface. For any one work session, do not switch between interfaces.

Descriptions of some tasks might mention task-related properties in the SKLMConfig.properties file. Use the graphical user interface, command-line interface, or REST interface to change these properties.

2. On the LTO Key and Device Management, you can add, modify, or delete a key, a key group, or drive.

You might do these administrative tasks:

- Refresh the list.

Click the refresh icon  to refresh items in the table.

- Add

Click **Add**. Alternatively, you can select a step-by-step process to create key groups, and drives.

- Key group

On the **Create Key Group** dialog, specify the required information such as the key group name. You might also specify that this group serves keys as the default key group. There can be only one default key group. Then, click **Create Key Group**. Your role must have a permission to the create action and a permission to the appropriate device group.

- Tape drive

On the Add Tape Drive dialog, type the drive serial number and other information. Then, click **Add Tape Drive**. Your role must have a permission to the create action and a permission to the appropriate device group.

- Use step by step process for key groups, keys, and drive creation

On the Step1: Create Key Groups and Step2: Identify Drives pages, enter the necessary information, and click the appropriate button to complete the task.

A success indicator varies, showing a key group or device.

- **Modify**

To change a key group, key, or drive, select a key group, key, or drive, and then click **Modify**. Alternatively, right-click the selected key group, key, or drive. Then, click **Modify**.

- **Key Group**

Specify changes on the Modify Key Group dialog. Then, click **Modify Key Group**. Your role must have a permission to the modify action and a permission to the appropriate device group.

- **Key**

Specify changes on the Modify Key Membership dialog. Then, click **Modify Key Membership**. Your role must have a permission to the modify action and a permission to the appropriate device group.

- **Tape drive**

Specify changes on the Modify Tape Drive dialog. Then, click **Modify Tape Drive**. Your role must have a permission to the modify action and a permission to the appropriate device group.

A success indicator varies, showing a change in a column for the key group, key, or device. Changes to optional information such as the value of a drive description might not be provided in the table.

- **Delete**

To delete a key group, key, or drive, select a key, a key group, or drive, and then click **Delete**. Alternatively, right-click the selected key group, key, or drive. Then, click **Delete**.

- **Key group**

You cannot delete a key group that is associated with a device, or a key group that is marked as default. Deleting a populated key group *also deletes all the keys* in the key group.

To confirm deletion, click **OK**. Your role must have a permission to the delete action and a permission to the appropriate device group.

- **Key**

Deleting a key removes the key from any key group with which the key is associated. To confirm deletion, click **OK**. You cannot delete a key that is associated with a drive. Your role must have a permission to the delete action and a permission to the appropriate device group.

- **Tape drive**

Metadata for the drive that you delete, such as the drive serial number, is removed from the IBM Security Key Lifecycle Manager database. To confirm deletion, click **OK**. Your role must have a permission to the delete action and a permission to the appropriate device group.

A success indicator is the deletion of the key group, key, or device from the management table.

Adding a key or key group

You might add more keys or key groups for use with IBM Security Key Lifecycle Manager.

About this task

You can use the Create Key Group dialog. Alternatively, you might first use the **tklmGroupCreate** command, or **Group Create REST Service** to create a group to which you want to add keys, and then use the **tklmSecretKeyCreate** command or **Secret Key Create REST Service** to create one or more symmetric keys in the existing group. Your role must have a permission to the create action and a permission to the appropriate device group.

Before you begin, determine your site policy on the default key groups and naming for key prefixes.

Procedure

1. Navigate to the appropriate page or directory:
 - Graphical user interface:
 - a. Log on to the graphical user interface.
 - b. In the Key and Device Management section on Welcome page, select **LTO**.
 - c. Click **Go to > Manage keys and devices**.
 - d. Alternatively, right-click **LTO** and select **Manage keys and devices**.
 - e. On the management page for LTO, click **Add**.
 - f. Click **Key Group**.
 - Command-line interface:

In the *WAS_HOME/bin* directory, start a **wsadmin** session by using Jython. Log on to **wsadmin** with an authorized user ID, such as the SKLMAdmin user ID. For example, on Windows systems, navigate to the *drive:\Program Files (x86)\IBM\WebSphere\AppServer\bin* directory and type:

 - Windows systems:

```
wsadmin -username SKLMAdmin -password mypwd -lang jython
```
 - Systems such as AIX or Linux:

```
./wsadmin.sh -username SKLMAdmin -password mypwd -lang jython
```
2. Create a key or key group:
 - Graphical user interface
 - a. On the Create Key Group dialog, specify values for the required and optional parameters. For example, you might optionally specify that this key group is the default.
 - b. Click **Create Key Group**.
 - Command-line interface:
 - a. First, create a group to which you might add keys.

Type **tklmGroupCreate** to create a group of that has a type of key group. For example, type:

```
print AdminTask.tklmGroupCreate  
('[-name GROUP-myKeyGroup -type keygroup -usage LTO]')
```
 - b. Next, use the **tklmGroupList** command obtain the value of the uuid for the group that you created. For example, type:

```
print AdminTask.tklmGroupList  
('[-name GROUP-myKeyGroup -type keygroup -v y]')
```
 - c. Then, create a group of keys and store them in the group. For example, type:

```
print AdminTask.tklmSecretKeyCreate ('[-alias abc
-keyStoreName defaultKeyStore
-numOfKeys 10 -usage LTO
-keyGroupUuid KEYGROUP-316408ac-f433-4c11-92bc-0de46d05bee9]')
```

- REST interface:

- a. Create a group to which you might add keys by using **Group Create REST Service**.

For example, you can send the following HTTP request by using a REST client:

```
POST https://localhost:9080/SKLM/rest/v1/keygroups/newGroup
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth authId=139aeh34567m
{"usage":"LTO"}
```

- b. Use **Group List REST Service** to obtain the value of the uuid for the group that you created. For example, you can send the following HTTP request:

```
GET https://localhost:9080/SKLM/rest/v1/keygroups?name=newGroup
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth authId=139aeh34567m
Accept-Language : en
```

- c. Create a group of keys and store them in the group by using **Secret Key Create REST Service**. For example, you can send the following HTTP request:

```
POST https://localhost:9080/SKLM/rest/v1/keys
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth authId=139aeh34567m
{"alias":"abc","numOfKeys":"10","KEYGROUP-316408ac-f433-4c11-92bc-0de46d05bee9","usage":"LTO"}
```

3. A success indicator varies, depending on the interface:

- Graphical user interface:

The key group is listed as an item in the **Key Group** column.

- Command-line interface:

Completion messages indicate success.

- Rest interface:

The status code 200 OK indicates success.

What to do next

Back up new keys before the keys are served to devices. You might also associate key groups with specific devices.

Specifying a rollover key group

You might specify a key group for future use as the system default.

About this task

You can use the graphical user interface, **tklmKeyGroupDefaultRolloverAdd** command or **Key Group Default Rollover Add REST Service** to add a default key group rollover on a specific date to serve keys to a device group. Your role must have a permission to the create action and a permission to the appropriate device group.

Procedure

1. Navigate to the appropriate page or directory:
 - Graphical user interface:
 - a. Log on to the graphical user interface.
 - b. In the Key and Device Management section on Welcome page, select **LTO**.
 - c. Click **Go to > Manage default rollover**.
 - d. Alternatively, right-click **LTO** and select **Manage default rollover**.
 - Command-line interface:

In the *WAS_HOME/bin* directory, start a **wsadmin** session by using Jython. Log on to **wsadmin** with an authorized user ID, such as the SKLMAdmin user ID. For example, on Windows systems, navigate to the *drive:\Program Files (x86)\IBM\WebSphere\AppServer\bin* directory and type:

 - Windows systems:

```
wsadmin -username SKLMAdmin -password mypwd -lang jython
```
 - Systems such as AIX or Linux:

```
./wsadmin.sh -username SKLMAdmin -password mypwd -lang jython
```
2. Specify an existing key group to be a future system default.
 - Graphical user interface:
 - a. On the management page for LTO, click **Add**.
 - b. On the Add Future Write Default dialog, specify the required information.
 - c. Click **Add Future Write Default**.

Note:

- Do not specify two defaults for the same rollover date.
 - If a key group does not exist at the time of rollover, IBM Security Key Lifecycle Manager continues to use the current default key group.
 - You can add or delete table entries, but cannot modify an entry.
- Command-line interface:

Add a rollover key group. For example, type:

```
print AdminTask.tklmKeyGroupDefaultRolloverAdd  
(['-usage LTO -keyGroupName myLTOkeygroup  
-effectiveDate 2010-04-30'])
```
3. A success indicator varies, depending on the interface:
 - Graphical user interface:

The rollover key group is listed in the table of rollover key groups on the LTO management page.
 - Command-line interface:

A completion message indicates success.
 4. To delete a key group from the rollover table, your role must have permission to the delete action.
 - Graphical user interface:

Select a key group and click **Delete**.
 - Command-line interface:

Use the **tklmKeyGroupDefaultRolloverList** command to locate the Universal Unique Identifier for a key group. Your role must have a permission to the view action and a permission to the appropriate device group. Then, use

tklmKeyGroupDefaultRolloverDelete command to remove the key group from the rollover list. Your role must have a permission to the delete action and a permission to the appropriate device group.

For example, type:

```
print AdminTask.tklmKeyGroupDefaultRolloverList
('[-usage LT0]')
print AdminTask.tklmKeyGroupDefaultRolloverDelete
('[-uuid 201]')
```

Specifying that keys are used only once

You might specify that keys in a key group are used only once. For security reasons, for example, you might prevent additional use of previously used keys that are defined for a key group.

About this task

You can use the command-line interface and the **stopRoundRobinKeyGrps** property in the `SKLMConfig.properties` file. Your role must have a permission to the configure action. This property is not initially present in the property file unless you set its value to true. This property can only be set by using the command-line interface.

Important:

- Turning on this flag can cause key serving to stop if a key group is in use and the last key from the key group is served. Additional requests for a key from this group on a key serving write request cause an error and send an error code of 0xEE34 (NO_KEY_TO_SERVE) to the device. To enable successful processing of new key serving write requests, add new keys to the key group. Alternatively, you might specify use of a different key group that has available keys. Key serving read requests always succeed when the requested key exists.
- Use this property in an environment of strict government compliance and with FIPS 140. With the property on, you must actively monitor your key groups. Ensure that a key group does not run out of keys, causing the server to stop serving keys and the tape write request to fail.
- If you turn on this flag, do not turn off the flag. For example, if you turn on the flag, a key group does not serve previously used keys. If you turn off the flag, the next key in the group is served. After the last key in the group is served, the next key to be served is the first key in the group.
- When this option is set, do not separately assign individual key aliases that belong to a key group to devices.

Procedure

1. Navigate to the appropriate directory:

In the `WAS_HOME/bin` directory, start a **wsadmin** session by using Jython. Log on to **wsadmin** with an authorized user ID, such as the SKLMAdmin user ID. For example, on Windows systems, navigate to the `drive:\Program Files (x86)\IBM\WebSphere\AppServer\bin` directory and type:

- Windows systems:
`wsadmin -username SKLMAdmin -password mypwd -lang jython`
- Systems such as AIX or Linux:
`./wsadmin.sh -username SKLMAdmin -password mypwd -lang jython`

2. First, determine the current state of the property in the `SKLMConfig.properties` file. This property is not initially present in the property file unless you set its value to true.

- Command-line interface:

At a **wsadmin** prompt, type this Jython-formatted command:

```
print AdminTask.tklmConfigGetEntry  
  ('[-name stopRoundRobinKeyGrps]')
```

- REST interface:

Use **Get Single Config Property REST Service** to get the current value of the property. Send the following HTTP request:

```
GET https://localhost:9080/SKLM/rest/v1/configProperties/  
stopRoundRobinKeyGrps  
Content-Type: application/json  
Accept : application/json  
Authorization: SKLMAuth userAuthId=139aeh34567m  
Accept-Language : en
```

3. Change the state of the **stopRoundRobinKeyGrps** property to a value of true in the `SKLMConfig.properties` file.

- Command-line interface:

Type this Jython-formatted command:

```
print AdminTask.tklmConfigUpdateEntry ('[-name stopRoundRobinKeyGrps  
-value true]')
```

- REST interface:

Send the following HTTP request:

```
PUT https://localhost:9080/SKLM/rest/v1/configProperties  
Content-Type: application/json  
Accept: application/json  
Authorization: SKLMAuth authId=139aeh34567m  
Accept-Language: en  
{ "stopRoundRobinKeyGrps": "true"}
```

4. To determine success, retype the **tklmConfigGetEntry** command or use **Get Single Config Property REST Service**.

Additionally, on the Welcome page in the graphical user interface, you might observe a warning in the Action Items section. The section lists key groups with 10 percent or fewer available keys. Double-click an entry in this table to access the Modify Key Groups dialog, where you can add more keys for use by the group.

There is no other warning. The low key count warning applies to all key groups, including the key group that is specified as the default.

Modifying a key group

You might modify information about objects in a key group in the IBM Security Key Lifecycle Manager database.

About this task

You can use the Modify Key Group dialog. Alternatively, you can use the following commands or REST interfaces to modify objects in a key group in the IBM Security Key Lifecycle Manager database.

- **tklmGroupEntryAdd** and **tklmGroupEntryDelete**
- **Group Entry Add REST Service** and **Group Entry Delete REST Service**

Your role must have a permission to the modify action and a permission to the appropriate device group.

Before you begin, determine the changed information for the group, such as the number of more keys that you want to add to the group.

Procedure

1. Navigate to the appropriate page or directory:

- Graphical user interface:
 - a. Log on to the graphical user interface.
 - b. In the Key and Device Management section on Welcome page, select **LTO**.
 - c. Click **Go to > Manage keys and devices**.
 - d. Alternatively, right-click **LTO** and select **Manage keys and devices**.
 - e. On the management page for LTO, select a key group in the **Key Group** column.
 - f. Click **Modify**.
 - g. Alternatively, right-click a key group and then select **Modify**, or double-click a key group entry.
- Command-line interface:

In the `WAS_HOME/bin` directory, start a **wsadmin** session by using Jython. Log on to **wsadmin** with an authorized user ID, such as the SKLMAdmin user ID. For example, on Windows systems, navigate to the `drive:\Program Files (x86)\IBM\WebSphere\AppServer\bin` directory and type:

 - Windows systems:

```
wsadmin -username SKLMAdmin -password mypwd -lang jython
```
 - Systems such as AIX or Linux:

```
./wsadmin.sh -username SKLMAdmin -password mypwd -lang jython
```

2. Modify the key group information:

- Graphical user interface:
 - a. On the Modify Key Group dialog, change the appropriate fields. Your role must have specific permissions for each action. For example, to delete a key from the group, your role must have a permission to the delete action and a permission to the appropriate device group.
 - b. Click **Modify Key Group**.
- Command-line interface:

You might delete an object in a group, or add an object to a group.

 - Delete a key from the group. Your role must have a permission to the delete action and a permission to the appropriate device group. For example, type:

```
print AdminTask.tklmGroupEntryDelete('[-entry "{type key} {uuid KEY-a3ce9230-bef9-42bd-86b7-6d208ec119cf}" -name GROUP-myKeyGroup -type keygroup]')
```
 - Add the same key back into the group again. Your role must have a permission to the modify action and a permission to the appropriate device group. For example, type:

```
print AdminTask.tklmGroupEntryAdd('[-name GROUP-myKeyGroup -type keygroup -entry "{type key} {alias aaa000000000000000000000000} {keyStoreName defaultKeyStore}"']')
```
- REST interface:

To delete a key from the group, you can send the following HTTP request:

```
DELETE https://localhost:9080/SKLM/rest/v1/keygroups/KEY-a3ce9230-bef9-42bd-86b7-6d208ec119cf
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
```

To add the same key back into the group again, you can send the following HTTP request:

```
POST https://localhost:9080/SKLM/rest/v1/keygroupentry
Content-Type: application/json
Accept : application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
{"name": "GROUP-myKeyGroup", "entry": "KEY-a3ce9230-bef9-42bd-86b7-6d208ec119cf"}
```

3. A success indicator varies, depending on the interface:

- Graphical user interface:
For required fields, a column displays changed data. For optional fields, you might need to reopen the Modify Key Group dialog to see the changed values, and then click **Cancel**.
- Command-line interface:
A completion message indicates success.
- Rest interface:
The status code 200 OK indicates success.

What to do next

Next, you might use the LTO Key and Device Management page to associate the key group with specific devices.

Deleting a key or key group

You might delete a selected key or key group. You cannot delete a key or a key group that is associated with a device, or a key group that is marked as the default key group.

About this task

Delete keys only when the data protected by those keys is no longer needed. Deleting keys is like erasing the data. After keys are deleted, data that is protected by those keys is not retrievable.

You can use the **Delete** menu item. Alternatively, you can use the following commands or REST services to delete a key, or to delete the key group.

- **tklmKeyDelete** or **Delete Key REST Service**
- **tklmGroupDelete** or **Group Delete REST Service**

Your role must have a permission to the delete action and a permission to the appropriate device group.

Before you delete, carry out the following verifications:

- Key
Ensure that a backup exists of the keystore with the key that you intend to delete.
- Key group

If you use the command-line interface, obtain the uuid of the key group that you intend to delete. Verify that the key group is not currently associated with a device, and is not marked as a default key group.

Procedure

1. Navigate to the appropriate page or directory:

- Graphical user interface:
 - a. Log on to the graphical user interface.
 - b. In the Key and Device Management section on Welcome page, select **LTO**.
 - c. Click **Go to > Manage keys and devices**.
 - d. Alternatively, right-click **LTO** and select **Manage keys and devices**.
 - e. On the management for LTO, select a key or key group in the appropriate column.
 - f. Click **Delete**.
 - g. Alternatively, right-click a key or key group and then select **Delete**.
- Command-line interface:

In the `WAS_HOME/bin` directory, start a **wsadmin** session by using Jython. Log on to **wsadmin** with an authorized user ID, such as the SKLMAdmin user ID. For example, on Windows systems, navigate to the `drive:\Program Files (x86)\IBM\WebSphere\AppServer\bin` directory and type:

 - Windows systems:

```
wsadmin -username SKLMAdmin -password mypwd -lang jython
```
 - Systems such as AIX or Linux:

```
./wsadmin.sh -username SKLMAdmin -password mypwd -lang jython
```

2. Delete the key or key group:

- Graphical user interface:

On the Confirm dialog, read the confirmation message before you delete the key or key group. Verify that the correct key or key group was selected. For example, you might delete an empty key group. Deleting a populated key group *also deletes all the keys* in the key group. Deleting a key that belongs to a key group also removes the key from the group. Then, click **OK**.
- Command-line interface:
 - Key
Type `tklmKeyDelete` to delete a key. For example, to delete a key that is not currently associated with a device, first locate the key. You might use the **tklmKeyList** command to find the key that you want to delete. For example, type:

```
print AdminTask.tklmKeyList ('[-usage LTO  
-attributes "{state active}" -v y]')
```

Then, delete the key. For example, type:

```
print AdminTask.tklmKeyDelete ('[-alias aaa000000000000000000000  
-keyStoreName defaultKeyStore]')
```

Deleting a key deletes the key material from the database.
 - Key group
Type `tklmGroupDelete` to delete a key group. For example, you might delete an empty key group. Deleting a populated key group *also deletes all the keys* in the key group. For example, to delete a key group that is not currently associated with a device, type:

```
print AdminTask.tklmGroupDelete  
(['-uuid GROUP-7d588437-e725-48bf-a836-00a47df64e78'])
```

- REST interface:

- Key

Use **Delete Key REST Service** to delete a key. Use **List Key REST Service** to find the key that you want to delete. For example, you can send the following HTTP request:

```
GET https://localhost:9080/SKLM/rest/v1/keys  
Content-Type: application/json  
Accept : application/json  
Authorization: SKLMAuth userAuthId=139aeh34567m
```

Send the following HTTP request to delete the key:

```
DELETE https://localhost:9080/SKLM/rest/v1/keys/aaa000000000000000000000  
Content-Type: application/json  
Accept : application/json  
Authorization: SKLMAuth userAuthId=139aeh34567m
```

- Key Group

Use **Group Delete REST Service** to delete a key group. For example, you can send the following HTTP request:

```
https://localhost:9080/SKLM/rest/v1/keygroups/GROUP-7d588437-e725-  
48bf-a836-00a47df64e78  
Content-Type: application/json  
Accept: application/json  
Authorization: SKLMAuth userAuthId=139aeh34567m
```

3. A success indicator varies, depending on the interface:

- Graphical user interface:

The key or key group is removed from the table. The key material is deleted from the keystore.

- Command-line interface:

A completion message indicates success.

- Rest interface:

The status code 200 OK indicates success.

What to do next

Refresh the table to ensure that the key or key group is deleted. Back up the keystore to accurately reflect the change in keys. Back up the database to reflect the change in key groups.

Adding a drive

You might add a device such as a tape drive to the IBM Security Key Lifecycle Manager database.

About this task

You can use the Add Tape Drive dialog. Alternatively, you can use the **tklmDeviceAdd** command or **Device Add REST Service** to add a device. Your role must have a permission to the create action and a permission to the appropriate device group.

Before you begin, create the keys and key groups that you want to associate with the devices that you are about to identify. Additionally, obtain the tape drive serial number, and other description information. Determine whether the drive uses a

specific key group, or a system default key group.

Procedure

1. Navigate to the appropriate page or directory:

- Graphical user interface:
 - a. Log on to the graphical user interface.
 - b. In the Key and Device Management section on Welcome page, select **LTO**.
 - c. Click **Go to > Manage keys and devices**.
 - d. Alternatively, right-click **LTO** and select **Manage keys and devices**.
 - e. On the management page for LTO, click **Add**.
 - f. Click **Tape Drive**.
- Command-line interface:

In the *WAS_HOME/bin* directory, start a **wsadmin** session by using Jython. Log on to **wsadmin** with an authorized user ID, such as the SKLMAdmin user ID. For example, on Windows systems, navigate to the *drive:\Program Files (x86)\IBM\WebSphere\AppServer\bin* directory and type:

 - Windows systems:

```
wsadmin -username SKLMAdmin -password mypwd -lang jython
```
 - Systems such as AIX or Linux:

```
./wsadmin.sh -username SKLMAdmin -password mypwd -lang jython
```

2. Add a device:

- Graphical user interface:

On the Add Tape Drive dialog, type the required and optional information. Then, click **Add Tape Drive**.
- Command-line interface:

Type **tklmDeviceAdd** to add a device. You must specify the device group and serial number. For example, type:

```
print AdminTask.tklmDeviceAdd ('[-type LTO -serialNumber FAA49403AQJF  
-attributes "{worldwideName ABCdeF1234567890}  
{description salesDivisionDrive} {symAlias LTOKeyGroup1}"]')
```
- REST interface:

Use **Device Add REST Service** to add a device. For example, you can send the following HTTP request:

```
POST https://localhost:9080/SKLM/rest/v1/devices  
Content-Type: application/json  
Accept : application/json  
Authorization : SKLMAuth userAuthId=37ea1939-1374-4db7-84cd-14e399be2d20  
Accept-Language : en  
{"type":"LTO","serialNumber":"FAA49403AQJF","attributes":"worldwideName  
ABCdeF1234567890,description salesDivisionDrive"}
```

3. A success indicator varies, depending on the interface:

- Graphical user interface:

The device is added to the table.
- Command-line interface:

A completion message indicates success.
- Rest interface:

The status code 200 OK indicates success.

What to do next

Next, you might determine the status of the drive that you added.

Modifying a drive

You might modify information about a device such as a tape drive in the IBM Security Key Lifecycle Manager database. For example, you might update the description of the drive.

About this task

You can use the Modify Tape Drive dialog, the **tklmDeviceUpdate** command, or **Device Update REST Service** to update a device. Your role must have a permission to the modify action and a permission to the appropriate device group.

Before you begin, create the keys and key groups that you want to associate with the devices that you are about to modify. If you use the command-line interface, obtain the value of the uuid for the device that you intend to update.

Procedure

1. Navigate to the appropriate page or directory:
 - Graphical user interface:
 - a. Log on to the graphical user interface.
 - b. In the Key and Device Management section on Welcome page, select **LTO**.
 - c. Click **Go to > Manage keys and devices**.
 - d. Alternatively, right-click **LTO** and select **Manage keys and devices**.
 - e. On the management page for LTO, select a drive in the **Tape Drives** column.
 - f. click **Modify**.
 - g. Alternatively, right-click a drive and then select **Modify**, or double-click a drive entry.
 - Command-line interface:

In the *WAS_HOME/bin* directory, start a **wsadmin** session by using Jython. Log on to **wsadmin** with an authorized user ID, such as the SKLMAdmin user ID. For example, on Windows systems, navigate to the *drive:\Program Files (x86)\IBM\WebSphere\AppServer\bin* directory and type:

 - Windows systems:

```
wsadmin -username SKLMAdmin -password mypwd -lang jython
```
 - Systems such as AIX or Linux:

```
./wsadmin.sh -username SKLMAdmin -password mypwd -lang jython
```
2. Modify a device:
 - Graphical user interface:
 - a. On the Modify Tape Drive dialog, type the required and optional information.
 - b. Click **Modify Tape Drive**.
 - Command-line interface:

Type **tklmDeviceUpdate** to update a device. You must specify the device uuid and the attributes that change. For example, type:

```
print AdminTask.tklmDeviceList ('[-type lto]')
```

```
print AdminTask.tklmDeviceUpdate
('[-uuid DEVICE-44b123ad-5ed8-4934-8c84-64cb9e11d990
-attributes "{symAlias LTOKey000001} {description myLT0drive}"]')
```

- REST interface:

Use **Device Update REST Service** to update a device. You must specify the device uuid and the attributes that change. For example, you can send the following HTTP request:

```
GET https://localhost:9080/SKLM/rest/v1/devices?type=LTO
Content-Type: application/json
Accept : application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
Accept-Language : en

PUT https://localhost:9080/SKLM/rest/v1/devices
Content-Type: application/json
Accept : application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
{"uuid":"DEVICE-44b123ad-5ed8-4934-8c84-64cb9e11d990","attributes":
"symAlias LTOKey000001,description myLT0drive"}
```

3. A success indicator varies, depending on the interface:

- Graphical user interface:

The device information is changed in the table.

- Command-line interface:

A completion message indicates success.

- Rest interface:

The status code 200 OK indicates success.

What to do next

Next, you might verify that the changes are made. For optional fields, such as the description, you might want to run the **tklmDeviceList** command or **Device List REST Service** to determine whether the value is changed. Alternatively, reopen the Modify Tape Drive dialog.

Deleting a drive

You might delete a device such as a tape drive. Metadata for the device that you delete, such as the drive serial number, is removed from the IBM Security Key Lifecycle Manager database.

About this task

You can use the Delete menu item, the **tklmDeviceDelete** command, or **Device Delete REST Service** to delete a device. Your role must have a permission to the delete action and a permission to the appropriate device group.

Before you begin, ensure that a current backup exists for the IBM Security Key Lifecycle Manager database. If you use the command-line interface or REST interface, obtain the uuid of the device that you intend to delete.

Procedure

1. Navigate to the appropriate page or directory:

- Graphical user interface:

- a. Log on to the graphical user interface.

- b. In the Key and Device Management section on Welcome page, select LTO.

- c. Click **Go to > Manage keys and devices**.
 - d. Alternatively, right-click **LTO** and select **Manage keys and devices**.
 - e. On the management page for LTO, select a device.
 - f. click **Delete**.
 - g. Alternatively, right-click a drive and then select **Delete**.
 - Command-line interface:
In the *WAS_HOME/bin* directory, start a **wsadmin** session by using Jython. Log on to **wsadmin** with an authorized user ID, such as the SKLMAdmin user ID. For example, on Windows systems, navigate to the *drive:\Program Files (x86)\IBM\WebSphere\AppServer\bin* directory and type:
 - Windows systems:
`wsadmin -username SKLMAdmin -password mypwd -lang jython`
 - Systems such as AIX or Linux:
`./wsadmin.sh -username SKLMAdmin -password mypwd -lang jython`
2. Delete the device:
- Graphical user interface:
On the Confirm dialog, read the confirmation message before you delete the device. Metadata for the device that you delete, such as the drive serial number, is removed from the IBM Security Key Lifecycle Manager database. Click **OK**.
 - Command-line interface:
Type `tklmDeviceDelete` to delete a device. You must specify the uuid. For example, type:

```
print AdminTask.tklmDeviceList ('[-type lto]')
print AdminTask.tklmDeviceDelete
      '[-uuid DEVICE-74386920-148c-47b2-a1e2-d19194b315cf]')
```
 - REST interface:
Use **Device Delete REST Service** to delete a device. You must specify the device uuid. For example, you can send the following HTTP request by using a REST client:

```
GET https://localhost:9080/SKLM/rest/v1/devices?type=LTO
Content-Type: application/json
Accept : application/json
Authorization : SKLMAuth userAuthId=37ea1939-1374-4db7-84cd-14e399be2d20
Accept-Language : en

DELETE https://localhost:9080/SKLM/rest/v1/devices/DEVICE-74386920-148c-47b2-a1e2-d19194b315cf
Content-Type: application/json
Accept : application/json
Authorization : SKLMAuth userAuthId=37ea1939-1374-4db7-84cd-14e399be2d20
```
3. A success indicator varies, depending on the interface:
- Graphical user interface:
The device is removed from the table.
 - Command-line interface:
A completion message indicates success.
 - Rest interface:
The status code 200 OK indicates success.

3592 tape drive management

You can manage 3592 tape drives by using IBM Security Key Lifecycle Manager.

Descriptions of some steps describe alternatives by using the graphical user interface, command-line interface, or the REST interface. For any one work session, do not switch between interfaces.

Descriptions of some tasks might mention task-related properties in the `SKLMConfig.properties` file. Use the graphical user interface, command-line interface, or REST interface to change these properties.

Guided steps to create certificates and drives

When you first create certificates and drives, and later when you add more certificates or drives, IBM Security Key Lifecycle Manager provides a guided set of steps to complete the task.

Descriptions of some steps might mention command-line or REST interface alternatives to do the same task. In a guided set of tasks, use the graphical user interface to complete the tasks.

Creating a certificate or certificate request

As a first activity, you might create certificates or certificate requests for IBM Security Key Lifecycle Manager.

About this task

You can use the Create Certificate dialog. Alternatively, you can use any of the following commands or REST services to create certificates or certificate requests:

- **tklmCertCreate** or **tklmCertGenRequest**
- **Create Certificate REST Service** or **Certificate Generate Request REST Service**

Your role must have a permission to the create action and permission to the appropriate device group. To make this certificate the default, your role must have permission to the modify action.

If you additionally want to specify that a certificate is used as the system default or partner certificate, you can use the following commands or REST services:

- **tklmDeviceGroupAttributeList** and **tklmDeviceGroupAttributeUpdate**
- **Device Group Attribute List REST Service** and **Device Group Attribute Update REST Service**

These values were previously stored in the obsolete **drive.default.alias1** (for system default) or **drive.default.alias2** (for system partner) properties.

Before you begin, determine your site policy for the use of self-signed and certificates that are issued by a certificate authority (CA). You might want to create self-signed certificates for the test phase of your project. In advance, you might also request certificates from a certificate authority for the production phase.

Procedure

1. Navigate to the appropriate page or directory:
 - Graphical user interface:
 - a. Log on to the graphical user interface.
 - b. In the Key and Device Management section on Welcome page, select **3592**.
 - c. Click **Go to > Guided key and device creation**.

- d. Alternatively, right-click **3592** and select **Guided key and device creation**.
 - Command-line interface:

In the *WAS_HOME/bin* directory, start a **wsadmin** session by using Jython. Log on to **wsadmin** with an authorized user ID, such as the SKLMAdmin user ID. For example, on Windows systems, navigate to the *drive:\Program Files (x86)\IBM\WebSphere\AppServer\bin* directory and type:

 - Windows systems:


```
wsadmin -username SKLMAdmin -password mypwd -lang jython
```
 - Systems such as AIX or Linux:


```
./wsadmin.sh -username SKLMAdmin -password mypwd -lang jython
```
 - REST interface:
 - Open a REST client.
2. Create a certificate or request a certificate:
- Graphical user interface:
 - a. On the On Step 1: Create Certificates page, click **Create** on the **Certificates** table.
 - b. On the Create Certificate dialog, select either a self-signed certificate, or a certificate request for a third-party provider.
 - c. Specify values for the required and optional parameters. For example, you might optionally specify that the certificate is the default or the partner certificate.
 - d. Click **Create Certificate**.
 - Command-line interface:
 - Certificate

Type **tklmCertCreate** to create a certificate and a public and private key pair, and store the certificate in an existing keystore. For example, type:

```
print AdminTask.tklmCertCreate ('[-type selfsigned
  -alias sklmCertificate -cn sklm -ou sales -o myCompanyName
  -usage 3592 -country US -keyStoreName defaultKeyStore
  -validity 999]')
```
 - Certificate request

Type **tklmCertGenRequest** to create a PKCS #10 certificate request file. For example, type:

```
print AdminTask.tklmCertGenRequest('[-alias sklmCertificate1
  -cn sklm -ou marketing -o CompanyName -locality myLocation
  -country US -validity 999 -keyStoreName defaultKeyStore
  -fileName myCertRequest1.crt -usage 3592]')
```
 - REST interface:
 - Certificate
 - a. Obtain a unique user authentication identifier to access IBM Security Key Lifecycle Manager REST services. For more information about the authentication process, see Authentication process for REST services.
 - b. To invoke **Create Certificate REST Service**, send the HTTP POST request. Pass the user authentication identifier that you obtained in Step a along with the request message as shown in the following example.

```
POST https://localhost:9080/SKLM/rest/v1/certificates
Content-Type: application/json
Accept : application/json
Authorization: SKLMAuth authId=139aeh34567m
```



```
Accept-Language : en
{"type":"selfsigned","alias":"sklmCertificate1","cn":"sklm","ou":"sales",
"o":"myCompanyName","usage":"3592","country":"US","validity":"999", "
algorithm ":" RSA " }
```

– Certificate request

Use **Certificate Generate Request REST Service** to create a PKCS #10 certificate request file. For example, you can send the following HTTP request:

```
POST https://localhost:9080/SKLM/rest/v1/certificates
Content-Type: application/json
Accept : application/json
Authorization: SKLMAuth authId=139aeh34567m
{"type":"certreq","alias":"sklmCertificate1","cn":"sklm","ou":"sales","o":
"myCompanyName","usage":"3592","country":"US","validity":"999","fileName":
"myCertRequest1.crt","algorithm":"ECDSA"}
```

3. A success indicator varies, depending on the interface:

- Graphical user interface:

The certificate or certificate request is listed as an item in the **Certificates** table.

- Command-line interface:

A completion message indicates success.

- Rest interface:

The status code 200 OK indicates success.

What to do next

Back up new certificates before the certificates are served to devices. For a certificate request, the next step might be to import the signed certificate. You might go the next step to define specific devices, and associate certificates with the devices. Select **Step 2: Identify Drives** or click **Go to Next Step**.

For a 3592 device group, also specify values for the system default and partner certificates in the IBM Security Key Lifecycle Manager database. Use the **tklmDeviceGroupAttributeUpdate** command or **Device Group Attribute Update REST Service** to set these values.

Identifying drives

You might identify a 3592 tape drive for use with IBM Security Key Lifecycle Manager.

About this task

You can use the Add Tape Drives dialog, the **tklmDeviceAdd** command, or **Device Add REST Service** to add a device. Your role must have a permission to the create action and a permission to the appropriate device group.

Before you begin, create the certificates that you want to associate with the devices that you are about to identify.

Procedure

1. Navigate to the appropriate page or directory:

- Graphical user interface:
 - a. Log on to the graphical user interface.

- b. In the Key and Device Management section on Welcome page, select **3592**.
 - c. Click **Go to > Guided key and device creation**.
 - d. Alternatively, right-click **3592** and select **Guided key and device creation**.
- Command-line interface:

In the *WAS_HOME/bin* directory, start a **wsadmin** session by using Jython. Log on to **wsadmin** with an authorized user ID, such as the SKLMAdmin user ID. For example, on Windows systems, navigate to the *drive:\Program Files (x86)\IBM\WebSphere\AppServer\bin* directory and type:

 - Windows systems:


```
wsadmin -username SKLMAdmin -password mypwd -lang jython
```
 - Systems such as AIX or Linux:


```
./wsadmin.sh -username SKLMAdmin -password mypwd -lang jython
```
- REST interface:
 - Open a REST client.
2. Skip **Step 1: Create Certificates**. Click the **Go to Next Step** link or click **Step 2: Identify Drives**.
3. You might specify that IBM Security Key Lifecycle Manager holds new device requests for your approval. Your role must have a permission to the modify action and a permission to the appropriate device group.
 - Graphical user interface:

Select **Hold new device requests pending my approval**.
 - Command-line interface:

Use the **tklmDeviceGroupAttributeUpdate** command or **Device Group Attribute Update REST Service** to set the value of the **device.AutoPendingAutoDiscovery** attribute. For example, type:

```
print AdminTask.tklmDeviceGroupAttributeUpdate ('[-name 3592
-attributes "{device.AutoPendingAutoDiscovery 2}"]')
```
 - REST interface:
 - a. Obtain a unique user authentication identifier to access IBM Security Key Lifecycle Manager REST services. For more information about the authentication process, see Authentication process for REST services.
 - b. To invoke **Device Group Attribute Update REST Service** and to set the value of the **device.AutoPendingAutoDiscovery** attribute, send the HTTP PUT request. Pass the user authentication identifier that you obtained in Step a along with the request message as shown in the following example.

```
PUT https://localhost:9080/SKLM/rest/v1/deviceGroupAttributes
Content-Type: application/json
Accept : application/json
Authorization: SKLMAuth authId=139aeh34567m
{"name":"3592","attributes":"device.AutoPendingAutoDiscovery 2"}
```
4. Add a device:
 - Graphical user interface:
 - a. On the Step 2: Identify Drives page, in the **Devices** table, click **Add**.
 - b. On the Add Tape Drive dialog, type the required and optional information.
 - c. Click **Add Tape Drive**.
 - Command-line interface:

Type `tklmDeviceAdd` to add a device. You must specify the device group and serial number. For example, type:

```
print AdminTask.tklmDeviceAdd ('[-type 3592 -serialNumber CDA39403AQJF
-attributes "{worldwideName ABCdeF1234567890}
{description marketingDivisionDrive}
{aliasOne encryption_cert}"]')
```

- REST interface:

You can use **Device Add REST Service** to add a device. For example, you can send the following HTTP request:

```
POST https://localhost:9080/SKLM/rest/v1/devices
Content-Type: application/json
Accept : application/json
Authorization : SKLMAuth userAuthId=37ea1939-1374-4db7-84cd-14e399be2d20
Accept-Language : en
{"type":"3592","serialNumber":"CDA39403AQJF","attributes":{"worldwideName
ABCdeF1234567890,description marketingDivisionDrive"}}
```

5. A success indicator varies, depending on the interface:

- Graphical user interface:
The device is added to the **Devices** table.
- Command-line interface:
A completion message indicates success.
- Rest interface:
The status code 200 OK indicates success.

What to do next

Next, you might use the 3592 Key and Device Management page to view all certificates and devices.

Administering certificates and devices

To administer certificates and devices, you might want to determine their status. You might map their association, or add, modify, or delete specific certificates or devices.

About this task







Use the 3592 Key and Device Management page to map certificates to devices to determine status of items in the table. You might add, modify, or delete certificates or devices. Your role must have a permission to the view action and a permission to the appropriate device group.

Before you begin, examine the columns on the page, which provides buttons to add, modify, or delete a table item. To sort information, click a column header.

The table is organized in these areas:

- In left columns, information about certificates indicates the certificate name, whether the certificate is used as a system default or system partner, the expiration date, and status of the certificate.
- In right columns, information about drives indicates the drive name and whether the drive uses a system default as its default or partner certificate.
- Status icons indicate the status of a certificate.

Table 3. Status icons and their meanings

Icon	Description
	Certificate is in an active state.
	Certificate is in a compromised state.
	Certificate expires soon.
	Certificate is in an expired state.
	Certificate valid from future date, for migrated certificates with a future use time stamp.
	IBM Security Key Lifecycle Manager has third-party certificate requests that are waiting to be signed and imported.

Procedure

- Log on to the graphical user interface:
 - In the Key and Device Management section on Welcome page, select **3592**.
 - Click **Go to > Manage keys and devices**.
 - Alternatively, right-click **3592** and select **Manage keys and devices**.

Descriptions of some steps describe alternatives by using the graphical user interface, command-line interface, or the REST interface. For any one work session, do not switch between interfaces.

Descriptions of some tasks might mention task-related properties in the SKLMConfig.properties file. Use the graphical user interface, command-line interface, or REST interface to change these properties.
- On the 3592 Key and Device Management page, you can add, modify, or delete a certificate or drive. Additionally, you can monitor the status of certificates. You might do these administrative tasks:
 - Add

Click **Add**. Alternatively, you can select a step-by-step process to create certificates and drives.

 - Certificate

On the Create Certificate dialog, select the certificate type as either self-signed or from a third-party provider, and complete the required information. Then, click **Create Certificate**. Your role must have a permission to the create action and permission to the appropriate device group. To make this certificate the default, your role must have permission to the modify action.
 - Tape drive

On the Add Tape Drive dialog, type the drive information. Then, click **Add Tape Drive**. Your role must have a permission to the create action and a permission to the appropriate device group.
 - Use step by step process for certificate and drive creation

On the Step1: Create Certificates and Step2: Identify Drives pages, enter the necessary information.

A success indicator varies, showing a change in a column for the certificate or device.

- **Modify**

To change or delete a certificate or drive, select a certificate or drive, and then click **Modify**. Alternatively, right-click the selected certificate or drive. Then, click **Modify**, or double-click a certificate or device entry in the list.

- **Certificate**

Specify changes in the Modify Certificate dialog. Then, click **Modify Certificate**. Your role must have a permission to the modify action and a permission to the appropriate device group.

- **Tape drive**

Specify changes in the Modify Tape Drive dialog. Then, click **Modify Tape Drive**. Your role must have a permission to the modify action and a permission to the appropriate device group.

A success indicator varies, showing a change in a column for the certificate or device. Changes to some information, such as optional fields, might not be provided in the table.

- **Delete**

To delete a certificate or drive, highlight the entry in the table and click **Delete**. Alternatively, right-click the selected certificate or drive. Then, click **Delete**.

- **Certificate**

Ensure that you have a current backup of the keystore before you delete a certificate. Any tapes that are written by using this certificate become non-readable after the certificate is deleted. The certificate to be deleted can be in any state, such as active. Regardless of its state, you cannot delete a certificate that is associated with a device. You also cannot delete a certificate that is marked as either default or partner. Your role must have a permission to the delete action and a permission to the appropriate device group.

Deleting a certificate deletes the material from the database.

To confirm deletion, click **OK**.

- **Tape drive**

Metadata for the drive that you delete, such as the drive serial number, is removed from the IBM Security Key Lifecycle Manager database. To confirm deletion, click **OK**. Your role must have a permission to the delete action and a permission to the appropriate device group.

A success indicator is that the certificate or device is removed from the administration table.

Adding a certificate or certificate request

You might add more certificates or certificate requests for use with IBM Security Key Lifecycle Manager.

About this task

You can use the Create Certificate dialog. Alternatively, you can use any of the following commands or REST services to create certificates or certificate requests:

- **tklmCertCreate** or **tklmCertGenRequest**
- **Create Certificate REST Service** or **Certificate Generate Request REST Service**

Your role must have a permission to the create action and permission to the appropriate device group. To make this certificate the default, your role must have permission to the modify action.

Before you begin, determine your site policy on the use of self-signed and CA certificates. You might need to create self-signed certificates for the test phase of your project. In advance, you might also request certificates from a certificate authority for the production phase.

Procedure

1. Navigate to the appropriate page or directory:

- Graphical user interface:
 - a. Log on to the graphical user interface.
 - b. In the Key and Device Management section on Welcome page, select **3592**.
 - c. Click **Go to > Manage keys and devices**.
 - d. Alternatively, right-click **3592** and select **Manage keys and devices**.
 - e. On the management page for 3592, click **Add**.
 - f. Click **Certificate**.
- Command-line interface:

In the *WAS_HOME/bin* directory, start a **wsadmin** session by using Jython. Log on to **wsadmin** with an authorized user ID, such as the SKLMAdmin user ID. For example, on Windows systems, navigate to the *drive:\Program Files (x86)\IBM\WebSphere\AppServer\bin* directory and type:

 - Windows systems:

```
wsadmin -username SKLMAdmin -password mypwd -lang jython
```
 - Systems such as AIX or Linux:

```
./wsadmin.sh -username SKLMAdmin -password mypwd -lang jython
```

2. Create a certificate or request a certificate:

- Graphical user interface:
 - a. On the Create Certificate dialog, select either a self-signed certificate, or a certificate request for a third-party provider.
 - b. Specify values for the required and optional parameters. For example, you might optionally specify that this certificate is the default or the partner certificate. Then, click **Create Certificate**.
- Command-line interface:
 - Certificate:

Type **tklmCertCreate** to create a certificate and a public and private key pair, and store the certificate in an existing keystore. For example, type:

```
print AdminTask.tklmCertCreate ('[-type selfsigned  
-alias sklmCertificate -cn sklm -ou sales -o myCompanyName  
-usage 3592 -country US -keyStoreName defaultKeyStore  
-validity 999]')
```
 - Certificate request:

Type **tklmCertGenRequest** to create a PKCS #10 certificate request file. For example, type:

```
print AdminTask.tklmCertGenRequest('[-alias sklmCertificate1  
-cn sklm -ou marketing -o CompanyName -locality myLocation  
-country US -validity 999 -keyStoreName defaultKeyStore  
-fileName myCertRequest1.crt -usage 3592]')
```
- REST interface:

- Certificate

Use **Create Certificate REST Service** to create a certificate and a public and private key pair, and store the certificate in an existing keystore. For example, you can send the following HTTP request by using a REST client:

```
POST https://localhost:9080/SKLM/rest/v1/certificates
Content-Type: application/json
Accept : application/json
Authorization: SKLMAuth authId=139aeh34567m
Accept-Language : en
{"type":"selfsigned","alias":"sklmCertificate1","cn":"sklm","ou":
"sales","o":"myCompanyName","usage":"3592","country":"US","validity":
"999", "algorithm " : " RSA " }
```

- Certificate request

Use **Certificate Generate Request REST Service** to create a PKCS #10 certificate request file. For example, you can send the following HTTP request by using a REST client:

```
POST https://localhost:9080/SKLM/rest/v1/certificates
Content-Type: application/json
Accept : application/json
Authorization: SKLMAuth authId=139aeh34567m
{"type":"certreq","alias":"sklmCertificate1","cn":"sklm","ou":
"sales","o":"myCompanyName","usage":"3592","country":"US","validity":
"999","fileName":"myCertRequest1.crt","algorithm":"ECDSA"}
```

3. A success indicator varies, depending on the interface:

- Graphical user interface:

The certificate or certificate request appears as an item in the **Certificates** listing.

- Command-line interface:

A completion message indicates success.

- Rest interface:

The status code 200 OK indicates success.

What to do next

Your next action depends on whether you created a certificate or a certificate request.

- Certificate:

Back up new certificates before the certificates are served to devices. You might associate a certificate with a specific device.

- Certificate request:

Manually send the certificate request to a certificate authority. When the signed certificate returns, import the certificate by using a pending action item on the Welcome panel, or by using the **tklmCertImport** command or **Certificate Import REST Service**. When the import completes, back up the certificate to enable serving the certificate to a device.

Specifying a rollover certificate

You might specify a certificate for future use as the system default or system partner certificate.

About this task

You can use the graphical user interface, **tklmCertDefaultRolloverAdd** command, or **Cert Default Rollover Add REST Service** to add a default certificate rollover for a specific date and device group. Your role must have a permission to the create action and a permission to the appropriate device group.

Procedure

1. Navigate to the appropriate page or directory:

- Graphical user interface:
 - a. Log on to the graphical user interface.
 - b. In the Key and Device Management section on Welcome page, select **3592**.
 - c. Click **Go to > Manage default rollover**.
 - d. Alternatively, right-click **3592** and select **Manage default rollover**.

- Command-line interface:

In the *WAS_HOME/bin* directory, start a **wsadmin** session by using Jython. Log on to **wsadmin** with an authorized user ID, such as the SKLMAdmin user ID. For example, on Windows systems, navigate to the *drive:\Program Files (x86)\IBM\WebSphere\AppServer\bin* directory and type:

- Windows systems:

```
wsadmin -username SKLMAdmin -password mypwd -lang jython
```

- Systems such as AIX or Linux:

```
./wsadmin.sh -username SKLMAdmin -password mypwd -lang jython
```

2. Specify an existing certificate for future use as a system default or system partner certificate.

- Graphical user interface:
 - a. On the management page for 3592, click **Add**.
 - b. On the Add Future Write Default dialog, specify the required information.
 - c. Click **Add Future Write Default**.

Note:

- Do not specify two defaults for the same rollover date.
- No validation occurs on whether the selected certificate is expired or expires at the time of the rollover.
- If a certificate does not exist at the time of rollover, IBM Security Key Lifecycle Manager continues to use the current default certificate.
- You can add or delete table entries, but cannot modify an entry.

- Command-line interface:

Add a rollover certificate. For example, type:

```
print AdminTask.tklmCertDefaultRolloverAdd  
(['-usage 3592 -alias tklmcert1  
-certDefaultType 1 -effectiveDate 2010-05-30'])
```

3. A success indicator varies, depending on the interface:

- Graphical user interface:

The certificate appears in the table of rollover certificates on the 3592 page.
- Command-line interface:

A completion message indicates success.

- Rest interface:
The status code 200 OK indicates success.
4. To delete a certificate from the rollover table:
 - Graphical user interface:
Select a certificate and click **Delete**. Your role must have a permission to the delete action. Read the warning message. Then, click **OK**.
 - Command-line interface:
Use the **tklmCertDefaultRolloverList** command to locate the Universal Unique Identifier for a certificate. Your role must have a permission to the view action and a permission to the appropriate device group. Then, use the **tklmCertDefaultRolloverDelete** command to remove the certificate from the rollover list. Your role must have a permission to the delete action and a permission to the appropriate device group. For example, type:


```
print AdminTask.tklmCertDefaultRolloverDelete
(['-uuid 101'])
```

The certificate is unmarked as a future system default or partner certificate, but is otherwise not changed or deleted.

Modifying a certificate

You might modify whether a certificate is used as the system default or system partner certificate.

About this task

You can use the Modify Certificate dialog to modify a certificate. Alternatively, you can use the following commands or REST services:

- **tklmCertUpdate** or **Certificate Update REST Service** to modify the state of certificates, such as trusted or compromised, and to modify certificate information.
- **tklmDeviceTypeAttributeUpdate** or **Device Type Attribute Update REST Service** to set the certificate as the system default or system partner certificate.

Your role must have a permission to the modify action and a permission to the appropriate device group.

Before you begin, determine the changed information for the certificate, such as a description, or whether you want to make the certificate the system default or system partner certificate. If you use the command-line interface, obtain the value of the uuid for the certificate.

Procedure

1. Navigate to the appropriate page or directory:
 - Graphical user interface:
 - a. Log on to the graphical user interface.
 - b. In the Key and Device Management section on Welcome page, select **3592**.
 - c. Click **Go to > Manage keys and devices**.
 - d. Alternatively, right-click **3592** and select **Manage keys and devices**.
 - e. On the management page for 3592, select a certificate in the **Certificates** column.
 - f. Click **Modify**.

- g. Alternatively, right-click a certificate and then select **Modify**, or double-click a certificate entry.
- Command-line interface:

In the *WAS_HOME/bin* directory, start a **wsadmin** session by using Jython. Log on to **wsadmin** with an authorized user ID, such as the SKLMAdmin user ID. For example, on Windows systems, navigate to the *drive:\Program Files (x86)\IBM\WebSphere\AppServer\bin* directory and type:

 - Windows systems:


```
wsadmin -username SKLMAdmin -password mypwd -lang jython
```
 - Systems such as AIX or Linux:


```
./wsadmin.sh -username SKLMAdmin -password mypwd -lang jython
```
2. Modify the certificate information:
 - Graphical user interface:

On the Modify Certificate dialog, change the appropriate fields. Then, click **Modify Certificate**.
 - Command-line interface:

Type `tklmCertList` to find a certificate and `tklmCertUpdate` to update a certificate. You must specify the uuid of the certificate and the changed attribute. For example, to change the description, type:

```
print AdminTask.tklmCertList('[-usage 3592
-attributes "{state active}" -v y]')

print AdminTask.tklmCertUpdate
('[-uuid CERTIFICATE-99fc36a-4ab6a0e12343
-usage 3592 -attributes "{information {new information}}"]')
```
 - REST interface:

Use **Certificate List REST Service** to find a certificate. For example, you can send the following HTTP request:

```
GET https://localhost:9080/SKLM/rest/v1/certificates?attributes=
state active
Content-Type: application/json
Accept: application/json
Authorization : SKLMAuth userAuthId=37ea1939-1374-4db7-84cd-14e399be2d20
Accept-Language : en
```

Use **Certificate Update REST Service** to update a certificate. For example, you can send the following HTTP request:

```
PUT https://localhost:9080/SKLM/rest/v1/certificates
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth authId=139aeh34567m
{"uuid":"CERTIFICATE-99fc36a-4ab6a0e12343","usage":
"3592","attributes":"information newinformation" }
```
3. A success indicator varies, depending on the interface:
 - Graphical user interface:

If you modified the system default or system partner setting, the change appears in the System Default/Partner column of the **Certificates** table.
 - Command-line interface:

A completion message indicates success.
 - Rest interface:

The status code 200 OK indicates success.

What to do next

Next, you might use the 3592 Key and Device Management page to associate certificates with specific devices.

Deleting a certificate

You might delete a selected certificate, which can be in any state, such as active. You cannot delete a certificate that is associated with a device, or a certificate that is marked as either a default or partner certificate. For example, you might delete an expired certificate.

About this task

Delete certificates only when the data protected by those certificates is no longer needed. Deleting certificates is like erasing the data. After certificates are deleted, data that is protected by those certificates is not retrievable.

You can use the Delete menu item or the **tklmCertDelete** command or **Delete Certificate REST Service** to delete a certificate. Your role must have a permission to the delete action and a permission to the appropriate device group.

Before you begin, ensure that a backup exists of the keystore with the certificate that you intend to delete. Verify that the certificate is not currently associated with a device, and that the certificate is not marked as either a default or partner certificate. Determine the current state of the certificate, and ensure that deleting a certificate in this state conforms with your site policies.

Deleting a certificate deletes the material from the database.

Procedure

1. Navigate to the appropriate page or directory:

- Graphical user interface:
 - a. Log on to the graphical user interface.
 - b. In the Key and Device Management section on Welcome page, select **3592**.
 - c. Click **Go to > Manage keys and devices**.
 - d. Alternatively, right-click **3592** and select **Manage keys and devices**.
 - e. On the management page for 3592, select a certificate in the **Certificates** column.
 - f. Click **Delete**.
 - g. Alternatively, right-click a certificate and then select **Delete**.
- Command-line interface:

In the *WAS_HOME/bin* directory, start a **wsadmin** session by using Jython. Log on to **wsadmin** with an authorized user ID, such as the SKLMAdmin user ID. For example, on Windows systems, navigate to the *drive:\Program Files (x86)\IBM\WebSphere\AppServer\bin* directory and type:

 - Windows systems:

```
wsadmin -username SKLMAdmin -password mypwd -lang jython
```
 - Systems such as AIX or Linux:

```
./wsadmin.sh -username SKLMAdmin -password mypwd -lang jython
```

2. Delete the certificate:

- Graphical user interface:

On the Confirm dialog, read the confirmation message to verify that the correct certificate was selected before you delete the certificate. Then, click **OK**.

- Command-line interface:

Type `tklmCertList` to find a certificate and `tklmCertDelete` to delete a certificate. You must specify the certificate alias and the keystore name. For example, to delete an expired certificate that is not currently associated with a device, type:

```
print AdminTask.tklmCertList('[-usage 3592
-attributes "{state active}" -v y]')

print AdminTask.tklmCertDelete ('[-alias mycertalias
-keyStoreName defaultKeyStore]')
```

- REST interface:

Use **Certificate List REST Service** to find a certificate and **Delete Certificate REST Service** to delete a certificate. For examples, you can send the following HTTP requests:

```
GET https://localhost:9080/SKLM/rest/v1/certificates?attributes=
state active
Content-Type: application/json
Accept: application/json
Authorization : SKLMAuth userAuthId=37ea1939-1374-4db7-84cd-14e399be2d20
Accept-Language : en

DELETE https://localhost:9080/SKLM/rest/v1/certificates/mycertalias
Content-Type: application/json
Accept : application/json
Authorization: SKLMAuth authId=139aeh34567m
Accept-Language : en
```

3. A success indicator varies, depending on the interface:

- Graphical user interface:

The certificate is removed from the Certificate table.

- Command-line interface:

A completion message indicates success.

- Rest interface:

The status code 200 OK indicates success.

What to do next

Next, you might back up the keystore again to accurately reflect the change in certificates.

Adding a drive

You might add a device such as a tape drive to the IBM Security Key Lifecycle Manager database.

About this task

the `tklmDeviceAdd` command

You can use the Add Tape Drive dialog. Alternatively, you can use the `tklmDeviceAdd` command or **Device Add REST Service** to add a device. Your role must have a permission to the create action and a permission to the appropriate device group.

Before you begin, create the certificates that you want to associate with the devices that you are about to identify. Additionally, obtain the tape drive serial number, and other description information. Determine whether the drive uses a specific certificate, or a system default certificate.

Procedure

1. Navigate to the appropriate page or directory:

- Graphical user interface:
 - a. Log on to the graphical user interface.
 - b. In the Key and Device Management section on Welcome page, select **3592**.
 - c. Click **Go to > Manage keys and devices**.
 - d. Alternatively, right-click **3592** and select **Manage keys and devices**.
 - e. On the management page for 3592, click **Add**.
 - f. Click **Tape Drive**.
- Command-line interface:

In the *WAS_HOME/bin* directory, start a **wsadmin** session by using Jython. Log on to **wsadmin** with an authorized user ID, such as the SKLMAdmin user ID. For example, on Windows systems, navigate to the *drive:\Program Files (x86)\IBM\WebSphere\AppServer\bin* directory and type:

 - Windows systems:

```
wsadmin -username SKLMAdmin -password mypwd -lang jython
```
 - Systems such as AIX or Linux:

```
./wsadmin.sh -username SKLMAdmin -password mypwd -lang jython
```

2. Add a device:

- Graphical user interface:

On the Add Tape Drive dialog, type the required and optional information. Then, click **Add Tape Drive**.
- Command-line interface:

Type **tklmDeviceAdd** to add a device. You must specify the device group and serial number. For example, type:

```
print AdminTask.tklmDeviceAdd ('[-type 3592 -serialNumber CDA39403AQJF  
-attributes "{worldwideName ABCdeF1234567890}  
{description marketingDivisionDrive}  
{aliasOne encryption_cert}"]')
```
- REST interface:

Use **Device Add REST Service** to add a device. For example, you can send the following HTTP request:

```
POST https://localhost:9080/SKLM/rest/v1/devices  
Content-Type: application/json  
Accept : application/json  
Authorization : SKLMAuth userAuthId=37ea1939-1374-4db7-84cd-14e399be2d20  
Accept-Language : en  
{ "type": "3592", "serialNumber": "CDA39403AQJF", "attributes": "worldwideName  
ABCdeF1234567890,description salesDivisionDrive" }
```

3. A success indicator varies, depending on the interface:

- Graphical user interface:

The device is added to the table.
- Command-line interface:

A completion message indicates success.
- Rest interface:

The status code 200 OK indicates success.

What to do next

Next, you might determine the status of the drive that you added.

Modifying a drive

You might modify information about a device such as a tape drive in the IBM Security Key Lifecycle Manager database. For example, you might update the specification for a partner certificate that the drive uses, or specify an alternate device group within the same device family.

About this task

You can use the Modify Tape Drive dialog. Alternatively, you can use the **tklmDeviceUpdate** command or **Device Update REST Service** to update a device, or specify an alternate device group within the same device family. Your role must have a permission to the modify action and a permission to the appropriate device group.

Before you begin, create the certificates that you need to associate with the devices that you are about to modify. If you use the command-line interface, obtain the value of the uuid for the device that you intend to update. Also, obtain the alias of any certificate that is associated with the drive.

Procedure

1. Navigate to the appropriate page or directory:
 - Graphical user interface:
 - a. Log on to the graphical user interface.
 - b. In the Key and Device Management section on Welcome page, select **3592**.
 - c. Click **Go to > Manage keys and devices**.
 - d. Alternatively, right-click **3592** and select **Manage keys and devices**.
 - e. On the management page for 3592, select a drive in the **Tape Drives** column.
 - f. click **Modify**.
 - g. Alternatively, right-click a drive and then select **Modify**, or double-click a drive entry.
 - Command-line interface:

In the *WAS_HOME/bin* directory, start a **wsadmin** session by using Jython. Log on to **wsadmin** with an authorized user ID, such as the SKLMAdmin user ID. For example, on Windows systems, navigate to the *drive:\Program Files (x86)\IBM\WebSphere\AppServer\bin* directory and type:

 - Windows systems:

```
wsadmin -username SKLMAdmin -password mypwd -lang jython
```
 - Systems such as AIX or Linux:

```
./wsadmin.sh -username SKLMAdmin -password mypwd -lang jython
```
2. Modify a device:
 - Graphical user interface:

In the Modify Tape Drive dialog, type the required and optional information. Then, click **Modify Tape Drive**.

- Command-line interface:

Type `tklmDeviceList` to locate a device and `tklmDeviceUpdate` to update a device. You must specify the device uuid and the attributes that change. For example, type:

```
print AdminTask.tklmDeviceList ('[-type 3592]')
print AdminTask.tklmDeviceUpdate
  ('[-uuid DEVICE-64c588ad-5ed8-4934-8c84-64cb9e11d990
   -attributes "{aliasTwo myPartner99}"]')
```

- REST interface:

Use **Device List REST Service** to locate a device and **Device Update REST Service** to update a device. For example, you can send the following HTTP requests:

```
GET https://localhost:9080/SKLM/rest/v1/devices?type=3592
Content-Type: application/json
Accept : application/json
Authorization : SKLMAuth userAuthId=37ea1939-1374-4db7-84cd-14e399be2d20
Accept-Language : en

PUT https://localhost:9080/SKLM/rest/v1/devices
Content-Type: application/json
Accept : application/json
Authorization: SKLMAuth authId=139aeh34567m
{"uuid":"DEVICE-64c588ad-5ed8-4934-8c84-64cb9e11d990","attributes":
 "aliasTwo myPartner99"}
```

3. A success indicator varies, depending on the interface:

- Graphical user interface:

The device information is changed in the table.

- Command-line interface:

A completion message indicates success.

- Rest interface:

The status code 200 OK indicates success.

What to do next

Next, you might verify that the changes are made. For optional fields, such as the description, you might want to run the `tklmDeviceList` command or **Device List REST Service** to determine whether the value is changed. Alternatively, reopen the Modify Tape Drive dialog.

Deleting a drive

You might delete a device such as a tape drive. Metadata for the drive that you delete, such as the drive serial number, is removed from the IBM Security Key Lifecycle Manager database.

About this task

You can use the Delete menu item, the `tklmDeviceDelete` command, or **Device Delete REST Service** to delete a device. Your role must have a permission to the delete action and a permission to the appropriate device group.

Before you begin, ensure that a current backup exists for the certificates and devices at your site. Obtain the uuid of the device you intend to delete.

Procedure

1. Navigate to the appropriate page or directory:

- Graphical user interface:
 - a. Log on to the graphical user interface.
 - b. In the Key and Device Management section on Welcome page, select **3592**.
 - c. Click **Go to > Manage keys and devices**.
 - d. Alternatively, right-click **3592** and select **Manage keys and devices**.
 - e. On the management page for 3592, select a device.
 - f. click **Delete**.
 - g. Alternatively, right-click a drive and then select **Delete**.
- Command-line interface:

In the *WAS_HOME/bin* directory, start a **wsadmin** session by using Jython. Log on to **wsadmin** with an authorized user ID, such as the SKLMAdmin user ID. For example, on Windows systems, navigate to the *drive:\Program Files (x86)\IBM\WebSphere\AppServer\bin* directory and type:

 - Windows systems:

```
wsadmin -username SKLMAdmin -password mypwd -lang jython
```
 - Systems such as AIX or Linux:

```
./wsadmin.sh -username SKLMAdmin -password mypwd -lang jython
```

2. Delete the device:

- Graphical user interface:

On the Confirm dialog, read the confirmation message to verify that the correct device was selected before you delete the device. Metadata for the drive that you delete, such as the drive serial number, is removed from the IBM Security Key Lifecycle Manager database.

Then, click **OK**.
- Command-line interface:

Type `tklmDeviceList` to locate a device and `tklmDeviceDelete` to delete a device. You must specify the uuid. For example, type:

```
print AdminTask.tklmDeviceList ('[-type 3592]')  
print AdminTask.tklmDeviceDelete  
('[-uuid DEVICE-74386920-148c-47b2-a1e2-d19194b315cf]')
```
- REST interface:

Use **Device List REST Service** to locate a device and **Device Delete REST Service** to delete a device. For example, you can send the following HTTP requests:

```
GET https://localhost:9080/SKLM/rest/v1/devices?type=3592  
Content-Type: application/json  
Accept : application/json  
Authorization : SKLMAuth userAuthId=37ea1939-1374-4db7-84cd-14e399be2d20  
Accept-Language : en  
  
DELETE https://localhost:9080/SKLM/rest/v1/devices/DEVICE-74386920-148c-47b2-a1e2-d19194b315cf  
Content-Type: application/json  
Authorization: SKLMAuth authId=139aeh34567m  
Accept : application/json
```

3. A success indicator varies, depending on the interface:

- Graphical user interface:

The device is removed from the table.

- Command-line interface:
A completion message indicates success.
- Rest interface:
The status code 200 OK indicates success.

DS8000 storage image management

You can manage DS8000 storage images by using IBM Security Key Lifecycle Manager.

Descriptions of some steps describe alternatives by using the graphical user interface, command-line interface, or the REST interface. For any one work session, do not switch between interfaces.

Descriptions of some tasks might mention task-related properties in the SKLMConfig.properties file. Use the graphical user interface, command-line interface, or REST interface to change these properties.

Guided steps to create storage images and image certificates

When you create or add storage images and image certificates, IBM Security Key Lifecycle Manager provides a guided set of steps to complete the task.

Descriptions of some steps might mention command-line alternatives to do the same task. In a guided set of tasks, use the graphical user interface to complete the tasks.

Creating an image certificate or certificate request

As a first activity, you might create image certificates or certificate requests for IBM Security Key Lifecycle Manager.

About this task

You can use the Create Certificate dialog. Alternatively, you can use any of the following commands or REST services to create certificates or certificate requests:

- **tklmCertCreate** or **tklmCertGenRequest**
- **Create Certificate REST Service** or **Certificate Generate Request REST Service**

Your role must have a permission to the create action and permission to the appropriate device group. To make this certificate the default, your role must have permission to the modify action.

Procedure

1. Navigate to the appropriate page or directory:
 - Graphical user interface:
 - a. Log on to the graphical user interface.
 - b. In the Key and Device Management section on Welcome page, select **DS8000**.
 - c. Click **Go to > Guided key and device creation**.
 - d. Alternatively, right-click **DS8000** and select **Guided key and device creation**.
 - Command-line interface:

In the *WAS_HOME/bin* directory, start a **wsadmin** session by using Jython. Log on to **wsadmin** with an authorized user ID, such as the SKLMAdmin user ID. For example, on Windows systems, navigate to the *drive:\Program Files (x86)\IBM\WebSphere\AppServer\bin* directory and type:

- Windows systems:

```
wsadmin -username SKLMAdmin -password mypwd -lang jython
```

- Systems such as AIX or Linux:

```
./wsadmin.sh -username SKLMAdmin -password mypwd -lang jython
```

- REST interface:

- Open a REST client.

2. Create an image certificate or request a certificate:

- Graphical user interface:

- a. On the On Step 1: Create Certificates page, click **Create** on the **Certificates** table.
- b. On the Create Certificate dialog, select either a self-signed certificate, or a certificate request for a third-party provider.
- c. Specify values for the required and optional parameters.
- d. Click **Create Certificate**.

- Command-line interface:

- Certificate

Type `tklmCertCreate` to create a certificate and a public and private key pair, and store the certificate in an existing keystore. For example, type:

```
print AdminTask.tklmCertCreate ('[-type selfsigned
    -alias sklmCertificate -cn sklm -ou sales -o myCompanyName
    -usage DS8000 -country US -keyStoreName defaultKeyStore
    -validity 999]')
```

- Certificate request

Type `tklmCertGenRequest` to create a PKCS #10 certificate request file. For example, type:

```
print AdminTask.tklmCertGenRequest('[-alias sklmCertificate3
    -cn sklm -ou sales -o myCompanyName -locality myLocation
    -country US -validity 999 -keyStoreName defaultKeyStore
    -fileName myCertRequest3.crt -usage DS8000]')
```

- REST interface:

- Certificate

- a. Obtain a unique user authentication identifier to access IBM Security Key Lifecycle Manager REST services. For more information about the authentication process, see Authentication process for REST services.
- b. To invoke **Create Certificate REST Service**, send the HTTP POST request. Pass the user authentication identifier that you obtained in Step a along with the request message as shown in the following example.

```
POST https://localhost:9080/SKLM/rest/v1/certificates
Content-Type: application/json
Accept : application/json
Authorization: SKLMAuth authId=139aeh34567m
Accept-Language : en
{"type":"selfsigned","alias":"sklmCertificate","cn":"sklm","ou":"sales",
"o":"myCompanyName","usage":"DS8000","country":"US","validity":"999",
"algorithm ":" RSA " }
```

- Certificate request

Use **Certificate Generate Request REST Service** to create a PKCS #10 certificate request file. For example, you can send the following HTTP request by using a REST client:

```
POST https://localhost:9080/SKLM/rest/v1/certificates
Content-Type: application/json
Accept : application/json
Authorization: SKLMAuth authId=139aeh34567m
{"type":"certreq","alias":"sklmCertificate3","cn":"sklm","ou":"sales","o":
"myCompanyName","usage":"DS8000","country":"US","validity":"999","fileName":
"myCertRequest1.crt","algorithm":"ECDSA"}
```

3. A success indicator varies, depending on the interface:
 - Graphical user interface:
The certificate or certificate request appears as an item in the **Certificates** table. Back up new certificates before the certificates are served to devices.
 - Command-line interface:
A completion message indicates success.
 - Rest interface:
The status code 200 OK indicates success.

What to do next

Next, you might go the next step to define specific storage images, and specify certificates for the storage images. Select **Step 2: Identify Images** or click **Go to Next Step**.

Identifying storage images

You might identify a storage image (device) for use with IBM Security Key Lifecycle Manager.

About this task

You can use the Add Storage Image dialog. Alternatively, you can use the **tklmDeviceAdd** command or **Device Add REST Service** to add a storage image.

Before you begin, create the image certificates that you want to associate with the storage images that you are about to identify.

Procedure

1. Navigate to the appropriate page or directory:
 - Graphical user interface:
Log on to the graphical user interface. From the navigation tree, click **IBM Security Key Lifecycle Manager > Welcome**. Scroll down the page to the key and device management section. In **Guided key and device creation**, select **DS8000**. Then, click **Go**.
 - a. Log on to the graphical user interface.
 - b. In the Key and Device Management section on Welcome page, select **DS8000**.
 - c. Click **Go to > Guided key and device creation**.
 - d. Alternatively, right-click **DS8000** and select **Guided key and device creation**.
 - Command-line interface:
In the **WAS_HOME/bin** directory, start a **wsadmin** session by using Jython. Log on to **wsadmin** with an authorized user ID, such as the SKLMAdmin user

- ID. For example, on Windows systems, navigate to the *drive:\Program Files (x86)\IBM\WebSphere\AppServer\bin* directory and type:
- Windows systems:


```
wsadmin -username SKLMAdmin -password mypwd -lang jython
```
 - Systems such as AIX or Linux:


```
./wsadmin.sh -username SKLMAdmin -password mypwd -lang jython
```
 - REST interface:
 - Open a REST client.
2. Skip **Step 1: Create Certificates**. Click the **Go to Next Step** link or click **Step 2: Identify Drives**.
 3. You might specify that all incoming devices are added to a pending list, but are not automatically served keys upon request. You must accept or reject a device in the pending devices list before IBM Security Key Lifecycle Manager serves keys to the device upon request. Your role must have a permission to the modify action and a permission to the appropriate device group.
 - Graphical user interface:

Select **Hold new device requests pending my approval**.
 - Command-line interface:

Use the **tklmDeviceGroupAttributeUpdate** command or **Device Group Attribute Update REST Service** to set the value of the **device.AutoPendingAutoDiscovery** attribute. For example, type:

```
print AdminTask.tklmDeviceGroupAttributeUpdate ('[-name DS8000
-attributes "{device.AutoPendingAutoDiscovery 2}"]')
```
 - REST interface:
 - a. Obtain a unique user authentication identifier to access IBM Security Key Lifecycle Manager REST services. For more information about the authentication process, see Authentication process for REST services.
 - b. To invoke **Device Group Attribute Update REST Service** and to set the value of the **device.AutoPendingAutoDiscovery** attribute, send the HTTP PUT request. Pass the user authentication identifier that you obtained in Step a along with the request message as shown in the following example.

```
PUT https://localhost:9080/SKLM/rest/v1/deviceGroupAttributes
Content-Type: application/json
Accept : application/json
Authorization: SKLMAuth authId=139aeh34567m
{"name":"DS8000","attributes":"device.AutoPendingAutoDiscovery 2"}
```
 4. Add a storage image:
 - Graphical user interface:
 - a. On the Step 2: Identify Images page, in the table, click **Add**.
 - b. On the Add Storage Image dialog, type the required and optional information.
 - c. Click **Add Storage Image**.
 - Command-line interface:

Type **tklmDeviceAdd** to add a storage image. You must specify the storage image type, the serial number, and an image certificate. For example, type:

```
print AdminTask.tklmDeviceAdd ('[-type DS8000 -serialNumber CCCB31403AFF
-attributes "{worldwideName ABCdeF1234567890}
{description salesDivisionDrive}
{aliasOne myimagecertificate}"]')
```
 - REST interface:

You can use **Device Add REST Service** to add a storage image. For example, you can send the following HTTP request by using a REST client:

```
POST https://localhost:9080/SKLM/rest/v1/devices
Content-Type: application/json
Accept : application/json
Authorization : SKLMAuth userAuthId=37ea1939-1374-4db7-84cd-14e399be2d20
Accept-Language : en
{"type":"DS8000","serialNumber":"CCCB31403AFF","attributes":{"worldwideName
ABCdeF1234567890,description marketingDivisionDrive"}}
```

5. A success indicator varies, depending on the interface:

- Graphical user interface:
The storage image is added to the table.
- Command-line interface:
A completion message indicates success.
- Rest interface:
The status code 200 OK indicates success.

What to do next

Next, you might import the signed certificate. Alternatively, use the Key and Device Management page to view all storage images and image certificates.

Administering storage images and image certificates

To administer storage images and image certificates, you might want to determine their status. You might map their association, or add, modify, or delete specific certificates or storage images.

About this task

Use the DS8000 Key and Device Management page to map image certificates to storage images and to determine status of items in the table. You might add, modify, or delete image certificates or storage images. Your role must have a permission to the view action and a permission to the appropriate device group.

Before you begin, examine the columns on the page, which provides buttons to add, modify, or delete a table item. To sort information, click a column header.

The table is organized in these areas:

- In left columns, information about certificates indicates the certificate name, the expiration date, and status of the certificate.
- In right columns, information about storage images indicates the storage image name and associated image certificate.
- Status icons indicate the status of a certificate.

Table 4. Status icons and their meanings







Icon	Description
	Certificate is in an active state.
	Certificate is in a compromised state.
	Certificate expires soon.

Table 4. Status icons and their meanings (continued)

Icon	Description
	Certificate is in an expired state.
	Certificate valid from future date, for migrated certificates with a future use time stamp.
	IBM Security Key Lifecycle Manager has third-party certificate requests that are waiting to be signed and imported.

Procedure

- Log on to the graphical user interface.
 - In the Key and Device Management section on Welcome page, select **DS8000**.
 - Click **Go to > Manage keys and devices**.
 - Alternatively, right-click **DS8000** and select **Manage keys and devices**.

Descriptions of some steps describe alternatives by using the graphical user interface, command-line interface, or the REST interface. For any one work session, do not switch between interfaces.

Descriptions of some tasks might mention task-related properties in the SKLMConfig.properties file. Use the graphical user interface, command-line interface, or REST interface to change these properties.
- On the DS8000 Key and Device Management page, you can add, modify, or delete a storage image or image certificate.
You might do these administrative tasks:
 - Add

Click **Add**. Alternatively, you can select a step-by-step process to create certificates and storage images.

 - Certificate

On the Create Certificate page, select the certificate type as either the self-signed or a request from a third-party provider, and complete the required information. Then, click **Create Certificate**. Your role must have a permission to the create action and permission to the appropriate device group. To make this certificate the default, your role must have permission to the modify action.
 - Storage image

On the Add Storage Image page, type the storage image information. Then, click **Add Storage Image**. Your role must have a permission to the create action and a permission to the appropriate device group.
 - Use step by step process for certificate and storage image creation

On the Step1: Create Certificates and Step2: Identify Images pages, enter the necessary information.

A success indicator varies, showing a change in a column for the certificate or storage image.
 - Modify

To change information about a storage image or view information about a certificate, select a certificate or storage image, and then click **Modify**. Alternatively, right-click the selected certificate or storage image. Then, click **Modify**, or double-click the certificate or storage image entry.

 - Certificate

View read-only information in the Modify Certificate page. Your role must have a permission to the modify action and a permission to the appropriate device group.

- Storage image

Specify changes in the Modify Storage Image page. Then, click **Modify Storage Image**. Your role must have a permission to the modify action and a permission to the appropriate device group.

A success indicator varies, showing a change in a column for the certificate or storage image. Changes to some information, such as optional fields, might not be provided in the table.

- Delete

To delete a certificate or storage image, verify that the correct certificate or storage image was selected. Then, click **Delete**. Alternatively, right-click the selected certificate or storage image. Then, click **Delete**.

- Certificate

Ensure that you have a current backup of the keystore before you delete a certificate. Any storage image that is written by using this certificate becomes non-readable after the certificate is deleted. The certificate to be deleted can be in any state, such as active. Regardless of its state, you cannot delete a certificate that is:

- Associated with a storage image.
- Marked by a DS8000 Turbo drive as a primary certificate for image or secondary certificate for image.

Deleting a certificate deletes the material from the database.

To confirm deletion, click **OK**. Your role must have a permission to the delete action and a permission to the appropriate device group.

- Storage image

Metadata for the storage image that you delete, such as the serial number, is removed from the IBM Security Key Lifecycle Manager database. To confirm deletion, click **OK**. Your role must have a permission to the delete action and a permission to the appropriate device group.

A success indicator is deletion of the certificate or storage image from the administration table.

Adding an image certificate or certificate request

You might add more image certificates or certificate requests for use with IBM Security Key Lifecycle Manager.

About this task

You can use the Create Certificate dialog. Alternatively, you can use any of the following commands or REST services to create certificates or certificate requests:

- **tklmCertCreate** or **tklmCertGenRequest**
- **Create Certificate REST Service** or **Certificate Generate Request REST Service**

Your role must have a permission to the create action and permission to the appropriate device group. To make this certificate the default, your role must have permission to the modify action.

Before you begin, determine your site policy on the use of certificates.

Procedure

1. Navigate to the appropriate page or directory:

- Graphical user interface:
 - a. Log on to the graphical user interface.
 - b. In the Key and Device Management section on Welcome page, select **DS8000**.
 - c. Click **Go to > Manage keys and devices**.
 - d. Alternatively, right-click **DS8000** and select **Manage keys and devices**.
 - e. On the management page for DS8000, click **Add**.
 - f. Click **Certificate**.
- Command-line interface:

In the *WAS_HOME/bin* directory, start a **wsadmin** session by using Jython. Log on to **wsadmin** with an authorized user ID, such as the SKLMAdmin user ID. For example, on Windows systems, navigate to the *drive:\Program Files (x86)\IBM\WebSphere\AppServer\bin* directory and type:

 - Windows systems:

```
wsadmin -username SKLMAdmin -password mypwd -lang jython
```
 - Systems such as AIX or Linux:

```
./wsadmin.sh -username SKLMAdmin -password mypwd -lang jython
```

2. Create a certificate or request a certificate:

- Graphical user interface:
 - a. On the Create Certificate page, select either a self-signed certificate, or a certificate request for a third-party provider.
 - b. Specify values for the required and optional parameters. Then, click **Create Certificate**.
- Command-line interface:
 - Certificate:

Type **tklmCertCreate** to create a certificate and a public and private key pair, and store the certificate in an existing keystore. For example, type:

```
print AdminTask.tklmCertCreate ('[-type selfsigned  
-alias sklmCertificate -cn sklm -ou sales -o myCompanyName  
-usage DS8000 -country US -keyStoreName defaultKeyStore  
-validity 999]')
```
 - Certificate request:

Type **tklmCertGenRequest** to create a PKCS #10 certificate request file. For example, type:

```
print AdminTask.tklmCertGenRequest('[-alias sklmCertificate3  
-cn sklm -ou sales -o myCompanyName -locality myLocation  
-country US -validity 999 -keyStoreName defaultKeyStore  
-fileName myCertRequest3.crt -usage DS8000]')
```
- REST interface:
 - Certificate
Use **Create Certificate REST Service** to create a certificate and a public and private key pair, and store the certificate in an existing keystore. For example, you can send the following HTTP request by using a REST client:

```
POST https://localhost:9080/SKLM/rest/v1/certificates  
Content-Type: application/json  
Accept : application/json  
Authorization: SKLMAuth authId=139aeh34567m
```



```
Accept-Language : en
{"type":"selfsigned","alias":"sklmCertificate","cn":"sklm","ou":
"sales","o":"myCompanyName","usage":"DS8000","country":"US","validity":
"999", "algorithm " : " RSA " }
```

– Certificate request

Use **Certificate Generate Request REST Service** to create a PKCS #10 certificate request file. For example, you can send the following HTTP request by using a REST client:

```
POST https://localhost:9080/SKLM/rest/v1/certificates
Content-Type: application/json
Accept : application/json
Authorization: SKLMAuth authId=139aeh34567m
{"type":"certreq","alias":"sklmCertificate3","cn":"sklm","ou":
"sales","o":"myCompanyName","usage":"DS8000","country":"US","validity":
"999","fileName":"myCertRequest3.crt","algorithm":"ECDSA"}
```

3. A success indicator varies, depending on the interface:

- Graphical user interface:

The certificate or certificate request is listed as an item in the **Certificates** listing. Back up new certificates before the certificates are served to devices.

- Command-line interface:

A completion message indicates success.

- Rest interface:

The status code 200 OK indicates success.

What to do next

Your next action depends on whether you created a certificate or a certificate request.

- Certificate:

You might associate a certificate with a specific storage image.

- Certificate request:

Manually send the certificate request to a certificate authority. When the signed certificate returns, import the certificate by using a pending action item on the Welcome panel, or by using the **tklmCertImport** command or **Certificate Import REST Service**.

Modifying an image certificate

You might use the graphical user interface to view read-only information about an image certificate in the IBM Security Key Lifecycle Manager database. Using the command-line interface or REST interface, you can change a limited number of attributes.

About this task

You can use the Modify Certificate dialog to modify a certificate. Alternatively, you can use the following commands or REST services:

- **tklmCertUpdate** or **Certificate Update REST Service** to modify the state of certificates, such as trusted or compromised, and to modify certificate information.
- **tklmDeviceTypeAttributeUpdate** or **Device Type Attribute Update REST Service** to set the certificate as the primary or secondary certificate.

Your role must have a permission to the modify action and a permission to the appropriate device group.

Note: IBM Security Key Lifecycle Manager database changes that you make are configured on the DS8000 Turbo drive when the drive contacts IBM Security Key Lifecycle Manager.

Procedure

1. Navigate to the appropriate page or directory:

- Graphical user interface:
 - a. Log on to the graphical user interface.
 - b. In the Key and Device Management section on Welcome page, select **DS8000**.
 - c. Click **Go to > Manage keys and devices**.
 - d. Alternatively, right-click **DS8000** and select **Manage keys and devices**.
 - e. On the management page for DS8000, select a certificate in the **Certificates** column.
 - f. Click **Modify**.
 - g. Alternatively, right-click a certificate and then select **Modify**, or double-click a certificate entry.
- Command-line interface:

In the *WAS_HOME/bin* directory, start a **wsadmin** session by using Jython. Log on to **wsadmin** with an authorized user ID, such as the SKLMAdmin user ID. For example, on Windows systems, navigate to the *drive:\Program Files (x86)\IBM\WebSphere\AppServer\bin* directory and type:

 - Windows systems:

```
wsadmin -username SKLMAdmin -password mypwd -lang jython
```
 - Systems such as AIX or Linux:

```
./wsadmin.sh -username SKLMAdmin -password mypwd -lang jython
```

2. View (graphical user interface) or modify (command-line interface) the certificate information:

- Graphical user interface:

On the Modify Certificate dialog, view the read-only fields.
- Command-line interface:

Type **tklmCertList** to find a certificate and **tklmCertUpdate** to update a certificate. You must specify the uuid of the certificate and the changed attribute. For example, to change the information, type:

```
print AdminTask.tklmCertList('[-usage DS8000
-attributes "{state active}" -v y]')

print AdminTask.tklmCertUpdate
('[-uuid CERTIFICATE-33fc26e-5fb5a0e66143
-usage DS8000 -attributes "{information {new information}}"]')
```
- REST interface:

Use **Certificate List REST Service** to find a certificate. For example, you can send the following HTTP request:

```
GET https://localhost:9080/SKLM/rest/v1/certificates?attributes=
state active
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth userAuthId=37ea1939-1374-4db7-84cd-14e399be2d20
Accept-Language: en
```

Use **Certificate Update REST Service** to update a certificate. For example, you can send the following HTTP request:

```
PUT https://localhost:9080/SKLM/rest/v1/certificates
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth authId=139aeh34567m
{"uuid":"CERTIFICATE-33fc26e-5fb5a0e66143","usage":
"DS8000","attributes":"information {newinformation}" }
```

3. A success indicator varies, depending on the interface:

- Graphical user interface:
A column displays read-only data.
- Command-line interface:
A completion message indicates success.
- Rest interface:
The status code 200 OK indicates success.

What to do next

Next, you might use the DS8000 Key and Device Management page to associate image certificates with specific storage images.

Deleting an image certificate

You might delete a selected image certificate, which can be in any state, such as active. You cannot delete a certificate that is associated with a storage image. You also cannot delete a certificate that is identified as the primary certificate for image or secondary certificate for image. For example, you might delete an expired certificate.

About this task

Delete certificates only when the data protected by those certificates is no longer needed. Deleting certificates is like erasing the data. After certificates are deleted, data that is protected by those certificates is not retrievable.

You can use the Delete menu item or you can use the **tklmcertdelete** command or **Delete Certificate REST Service** to delete a selected image certificate. Your role must have a permission to the delete action and a permission to the appropriate device group.

Before you begin, ensure that a backup exists of the keystore with the image certificate that you intend to delete. Verify that the certificate is not currently associated with a storage image. Determine the current state of the certificate, and ensure that deleting a certificate in this state conforms with your site policies.

Deleting a certificate deletes the material from the database.

Procedure

1. Navigate to the appropriate page or directory:
 - Graphical user interface:
 - a. Log on to the graphical user interface.
 - b. In the Key and Device Management section on Welcome page, select **DS8000**.
 - c. Click **Go to > Manage keys and devices**.
 - d. Alternatively, right-click **DS8000** and select **Manage keys and devices**.

- e. On the management page for DS8000, select a certificate in the **Certificates** column.
 - f. Click **Delete**.
 - g. Alternatively, right-click a certificate and then select **Delete**.
 - Command-line interface:
 In the *WAS_HOME/bin* directory, start a **wsadmin** session by using Jython. Log on to **wsadmin** with an authorized user ID, such as the SKLMAdmin user ID. For example, on Windows systems, navigate to the *drive:\Program Files (x86)\IBM\WebSphere\AppServer\bin* directory and type:
 - Windows systems:
`wsadmin -username SKLMAdmin -password mypwd -lang jython`
 - Systems such as AIX or Linux:
`./wsadmin.sh -username SKLMAdmin -password mypwd -lang jython`
2. Delete the certificate:
- Graphical user interface:
 On the Confirm dialog, read the confirmation message to verify that the correct certificate was selected before you delete the certificate. Then, click **OK**.
 - Command-line interface:
 Type `tklmCertList` to find a certificate and `tklmCertDelete` to delete a certificate. You must specify the certificate alias and the keystore name. For example, to delete an expired certificate that is not currently associated with a storage image, type:


```
print AdminTask.tklmCertList('[-usage DS8000 -v y]')
print AdminTask.tklmCertDelete ('[-alias mycertalias
-keyStoreName defaultKeyStore]')
```
 - REST interface:
 Use **Certificate List REST Service** to find a certificate and **Delete Certificate REST Service** to delete a certificate. For examples, you can send the following HTTP requests:


```
GET https://localhost:9080/SKLM/rest/v1/certificates?usage=DS8000
Content-Type: application/json
Accept: application/json
Authorization : SKLMAuth userAuthId=37ea1939-1374-4db7-84cd-14e399be2d20
Accept-Language : en

DELETE https://localhost:9080/SKLM/rest/v1/certificates/mycertalias
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth authId=139aeh34567m
Accept-Language: en
```
3. A success indicator varies, depending on the interface:
- Graphical user interface:
 The certificate is removed from the Certificate table.
 - Command-line interface:
 A completion message indicates success.
 - Rest interface:
 The status code 200 OK indicates success.

What to do next

Next, you might back up the keystore again to accurately reflect the change in certificates.

Adding a storage image

You might add a storage image to the IBM Security Key Lifecycle Manager database.

About this task

You can use the Add Storage Image dialog or you can use the **tklmDeviceAdd** command or **Device Add REST Service** to add a storage image. Your role must have a permission to the create action and a permission to the appropriate device group.

Before you begin, create the certificates that you want to associate with the storage images that you are about to identify. Additionally, obtain the storage image serial number, and other description information.

Procedure

1. Navigate to the appropriate page or directory:

- Graphical user interface:
 - a. Log on to the graphical user interface.
 - b. In the Key and Device Management section on Welcome page, select **DS8000**.
 - c. Click **Go to > Manage keys and devices**.
 - d. Alternatively, right-click **DS8000** and select **Manage keys and devices**.
 - e. On the management page for DS8000, click **Add**.
 - f. Click **Storage Image**.

- Command-line interface:

In the *WAS_HOME/bin* directory, start a **wsadmin** session by using Jython. Log on to **wsadmin** with an authorized user ID, such as the SKLMAdmin user ID. For example, on Windows systems, navigate to the *drive:\Program Files (x86)\IBM\WebSphere\AppServer\bin* directory and type:

- Windows systems:

```
wsadmin -username SKLMAdmin -password mypwd -lang jython
```

- Systems such as AIX or Linux:

```
./wsadmin.sh -username SKLMAdmin -password mypwd -lang jython
```

2. Add a storage image:

- Graphical user interface:

On the Add Storage Image dialog, type the required and optional information. Then, click **Add Storage Image**.

- Command-line interface:

Type **tklmDeviceAdd** to add a storage image. You must specify the storage image type, the serial number, and an image certificate. For example, type:

```
print AdminTask.tklmDeviceAdd ('[-type DS8000 -serialNumber CCCB31403AFF  
-attributes "{worldwideName ABCdeF1234567890}  
{description salesDivisionDrive}  
{aliasOne myimagecertificate}"]')
```

- REST interface:

Use **Device Add REST Service** to add a storage image. For example, you can send the following HTTP request:

```
POST https://localhost:9080/SKLM/rest/v1/devices
Content-Type: application/json
Accept : application/json
Authorization : SKLMAuth userAuthId=37ea1939-1374-4db7-84cd-14e399be2d20
Accept-Language : en
{"type":"DS8000","serialNumber":"CCCB31403AFF","attributes":"worldwideName
ABCdeF1234567890,description salesDivisionDrive"}
```

3. A success indicator varies, depending on the interface:

- Graphical user interface:
The storage image is added to the table.
- Command-line interface:
A completion message indicates success.
- Rest interface:
The status code 200 OK indicates success.

What to do next

Next, you might determine the status of the storage image that you added.

Modifying a storage image

You might modify information about a storage image in the IBM Security Key Lifecycle Manager database. For example, you might update the storage image description.

About this task

You can use the Modify Storage Image dialog or you can use the **tklmDeviceUpdate** command or **Device Update REST Service** to update a storage image. Your role must have a permission to the modify action and a permission to the appropriate device group.

Before you begin, create the certificates that you want to associate with the storage images that you are about to modify. If you use the command-line interface, obtain the value of the uuid for the storage image that you intend to update and the alias of any certificate that is associated with the storage image.

Procedure

1. Navigate to the appropriate page or directory:
 - Graphical user interface:
 - a. Log on to the graphical user interface.
 - b. In the Key and Device Management section on Welcome page, select **DS8000**.
 - c. Click **Go to > Manage keys and devices**.
 - d. Alternatively, right-click **DS8000** and select **Manage keys and devices**.
 - e. On the management page for DS8000, select a drive.
 - f. click **Modify**.
 - g. Alternatively, right-click a drive and then select **Modify**, or double-click a drive entry.
 - Command-line interface:

In the *WAS_HOME/bin* directory, start a **wsadmin** session by using Jython. Log on to **wsadmin** with an authorized user ID, such as the SKLMAdmin user ID. For example, on Windows systems, navigate to the *drive:\Program Files (x86)\IBM\WebSphere\AppServer\bin* directory and type:

- Windows systems:

```
wsadmin -username SKLMAdmin -password mypwd -lang jython
```

- Systems such as AIX or Linux:

```
./wsadmin.sh -username SKLMAdmin -password mypwd -lang jython
```

2. Modify a storage image:

- Graphical user interface:

In the Modify Storage Image dialog, type the changed information. Then, click **Modify Storage Image**.

- Command-line interface:

Type **tklmDeviceUpdate** to update a storage image. You must specify the storage image uuid and the attributes that change. For example, type:

```
print AdminTask.tklmDeviceUpdate
(['[-uuid DEVICE-15d499ad-3ad8-3333-8c84-64cb9e11d990
  -attributes "{description myDevice}"']')
```

- REST interface:

Use **Device Update REST Service** to update a storage image. For example, you can send the following HTTP request:

```
PUT https://localhost:9080/SKLM/rest/v1/devices
Content-Type: application/json
Accept : application/json
Authorization: SKLMAuth authId=139aeh34567m
{"uuid":"DEVICE-15d499ad-3ad8-3333-8c84-64cb9e11d990","attributes":
"description myDevice"}
```

3. A success indicator varies, depending on the interface:

- Graphical user interface:

The storage image information is changed in the table.

- Command-line interface:

A completion message indicates success.

- Rest interface:

The status code 200 OK indicates success.

Deleting a storage image

You might delete a storage image. Metadata for the storage image that you delete, such as the serial number, is removed from the IBM Security Key Lifecycle Manager database.

About this task

You can use the Delete menu item or you can use the **tklmDeviceDelete** command, or **Device Delete REST Service** to delete a storage image. Your role must have a permission to the delete action and a permission to the appropriate device group.

Before you begin, ensure that a current backup exists for the certificates and storage images at your site. If you use the command-line interface, obtain the uuid of the storage image that you intend to delete.

Procedure

1. Navigate to the appropriate page or directory:

- Graphical user interface:
 - a. Log on to the graphical user interface.
 - b. In the Key and Device Management section on Welcome page, select **DS8000**.
 - c. Click **Go to > Manage keys and devices**.
 - d. Alternatively, right-click **DS8000** and select **Manage keys and devices**.
 - e. On the management page for DS8000, select a device.
 - f. click **Delete**.
 - g. Alternatively, right-click a drive and then select **Delete**.
- Command-line interface:

In the *WAS_HOME/bin* directory, start a **wsadmin** session by using Jython. Log on to **wsadmin** with an authorized user ID, such as the SKLMAdmin user ID. For example, on Windows systems, navigate to the *drive:\Program Files (x86)\IBM\WebSphere\AppServer\bin* directory and type:

 - Windows systems:

```
wsadmin -username SKLMAdmin -password mypwd -lang jython
```
 - Systems such as AIX or Linux:

```
./wsadmin.sh -username SKLMAdmin -password mypwd -lang jython
```

2. Delete the storage image:

- Graphical user interface:

On the Confirm page, read the confirmation message to verify that the correct storage image was selected before you delete the storage image. Metadata for the storage image that you delete, such as the serial number, is removed from the IBM Security Key Lifecycle Manager database.

Then, click **OK**.
- Command-line interface:

Type `tklmDeviceList` to locate a device and `tklmDeviceDelete` to delete a storage image. You must specify the uuid. For example, type:

```
print AdminTask.tklmDeviceList ('[-type DS8000]')
print AdminTask.tklmDeviceDelete
      '[-uuid DEVICE-74386920-148c-47b2-a1e2-d19194b315cf]')
```
- REST interface:

Use **Device List REST Service** to locate a device and **Device Delete REST Service** to delete a storage image. For example, you can send the following HTTP requests:

```
GET https://localhost:9080/SKLM/rest/v1/devices?type=DS8000
Content-Type: application/json
Accept : application/json
Authorization : SKLMAuth userAuthId=37ea1939-1374-4db7-84cd-14e399be2d20
Accept-Language : en

DELETE https://localhost:9080/SKLM/rest/v1/devices/DEVICE-74386920-148c-47b2-a1e2-d19194b315cf
Content-Type: application/json
Accept : application/json
Authorization: SKLMAuth authId=139aeh34567m
```

3. A success indicator varies, depending on the interface:

- Graphical user interface:

The storage image is removed from the table.

- Command-line interface:
A completion message indicates success.
- Rest interface:
The status code 200 OK indicates success.

DS5000 management

You can manage DS5000 storage servers by using IBM Security Key Lifecycle Manager.

Administering devices, keys, and device associations

To administer DS5000 storage servers, you map a device to keys or machines.

About this task

Your role must have a permission to the view action and a permission to the appropriate device group. Use the DS5000 Key and Device Management page to add, modify, or delete a device, key, or association. These actions require more permissions.

Before you begin, examine the columns on the page, which provides buttons to add, modify, or delete a table item. To sort information, click a column header.

The table is organized in these information areas:

- Devices and any associated machines.
- Current key that the device uses and a description of the device.

Procedure

1. Log on to the graphical user interface.
 - a. In the Key and Device Management section on Welcome page, select **DS5000**.
 - b. Click **Go to > Manage keys and devices**.
 - c. Alternatively, right-click **DS5000** and select **Manage keys and devices**.


Descriptions of some steps describe alternatives by using the graphical user interface, command-line interface, or the REST interface. For any one work session, do not switch between interfaces.

Descriptions of some tasks might mention task-related properties in the SKLMConfig.properties file. Use the graphical user interface, command-line interface, or REST interface to change these properties.

2. You can add, modify, or delete a key, device, or machine association.

You might do these administrative tasks:

- Refresh the list.

Click the refresh icon  to refresh items in the table.

- Add

Click **Add**.

- Device

On the Add Device dialog, type the device serial number and other information. Then, click **Add Device**. Your role must have a permission to the create action and a permission to the appropriate device group.

- Keys

Select a device and then select **Add > More Keys**. On the Add Key dialog, specify the required information such as the number of keys to create, up to a maximum of 12 keys. Then, click **Add > More Keys**. Your role must have a permission to the create action and a permission to the appropriate device group.

- Association

When machine affinity is enabled by the **device.enableMachineAffinity** property, use the Add Association dialog to specify the required information such as the machine ID. Then, click **Add Association**. Your role must have a permission to the create action and a permission to the appropriate device group.

A success indicator varies, showing the addition of a device, keys, or association.

- Modify

To change a device or keys, select the device and then click **Modify**. Alternatively, right-click the selected device. Then, click one of the choices, such as **Modify Device**.

- Device

Specify changes on the Modify Device dialog. Then, click **Modify Device**. Your role must have a permission to the modify action and a permission to the appropriate device group.

- Keys

Select a key on the Modify Keys dialog. Then, click **Delete**. Your role must have a permission to the delete action and a permission to the appropriate device group.

A success indicator varies, showing a change in a column for the device or key.

- Delete

To delete a device, select the device, and then click **Delete**. Alternatively, right-click the selected device. Then, click **Delete**. Before you delete the device, use the **tklmMachineDeviceDelete** command to remove the association of a device from an existing machine identifier in the IBM Security Key Lifecycle Manager database.

Metadata for the device that you delete, such as the device serial number, is removed from the IBM Security Key Lifecycle Manager database. Key data is also removed. To confirm deletion, click **OK**. Your role must have a permission to the delete action and a permission to the appropriate device group.

A success indicator is deletion of the device from the table.

Adding a device

You might add a device to the IBM Security Key Lifecycle Manager database.

About this task

If machine affinity is enabled, adding a device requires that you also add a relationship between a device and a machine. Otherwise, keys are not served to the added device. Using machine affinity, you can set key serving for specific device and machine combinations.

You can use the Add Device dialog, the **tklmDeviceAdd** command, or **Device Add REST Service** to add a device. Your role must have a permission to the create action and a permission to the appropriate device group.

Procedure

1. Navigate to the appropriate page or directory:

- Graphical user interface:
 - a. Log on to the graphical user interface.
 - b. In the Key and Device Management section on Welcome page, select **DS5000**.
 - c. Click **Go to > Manage keys and devices**.
 - d. Alternatively, right-click **DS5000** and select **Manage keys and devices**.
 - e. On the management page for DS5000, click **Add**.
 - f. Click **Device**.
- Command-line interface:

In the *WAS_HOME/bin* directory, start a **wsadmin** session by using Jython. Log on to **wsadmin** with an authorized user ID, such as the SKLMAdmin user ID. For example, on Windows systems, navigate to the *drive:\Program Files (x86)\IBM\WebSphere\AppServer\bin* directory and type:

 - Windows systems:

```
wsadmin -username SKLMAdmin -password mypwd -lang jython
```
 - Systems such as AIX or Linux:

```
./wsadmin.sh -username SKLMAdmin -password mypwd -lang jython
```
- REST interface:
 - Open a REST client.

2. Add a device:

- Graphical user interface:

On the Add Device dialog, type the required and optional information. Then, click **Add Device**.
- Command-line interface:

Type `tklmDeviceAdd` to add a device. You must specify the device serial number and device group. For example, type:

```
print AdminTask.tklmDeviceAdd ('[-type DS5000 -serialNumber CDA39403AQJF  
-attributes "{worldwideName ABCdeF1234567890}  
{description marketingDivisionDrive}  
{keyPrefix AEF}  
{numberOfKeys 10}  
{deviceText abcdefghijklmnopqrst}  
{machineID 304238303030343700000000000000}"']')
```
- REST interface:
 - a. Obtain a unique user authentication identifier to access IBM Security Key Lifecycle Manager REST services. For more information about the authentication process, see *Authentication process for REST services*.
 - b. To invoke **Device Add REST Service**, send the HTTP POST request. Pass the user authentication identifier that you obtained in Step a along with the request message as shown in the following example.

```
POST https://localhost:9080/SKLM/rest/v1/devices  
Content-Type: application/json  
Accept : application/json  
Authorization : SKLMAuth userAuthId=37ea1939-1374-4db7-84cd-14e399be2d20  
Accept-Language : en  
{ "type": "DS5000", "serialNumber": "CDA39403AQJF", "attributes": "worldwideName  
ABCdeF1234567890,description marketingDivisionDrive" }
```

3. A success indicator varies, depending on the interface:

- Graphical user interface:

The device is added to the table.

- Command-line interface:
A completion message indicates success.
- Rest interface:
The status code 200 OK indicates success.

What to do next

Next, you might associate the device with a machine.

Modifying a device

You might modify information about a device in the IBM Security Key Lifecycle Manager database. For example, you might update the description of the drive.

About this task

You can use the Modify Device dialog. Alternatively, you can use the **tklmDeviceUpdate** command or **Device Update REST Service** to update a device. Your role must have a permission to the modify action and a permission to the appropriate device group.

Before you begin, if you use the command-line or REST interface, obtain the value of the uuid for the device that you intend to update.

Procedure

1. Navigate to the appropriate page or directory:
 - Graphical user interface:
 - a. Log on to the graphical user interface.
 - b. In the Key and Device Management section on Welcome page, select **DS5000**.
 - c. Click **Go to > Manage keys and devices**.
 - d. Alternatively, right-click **DS5000** and select **Manage keys and devices**.
 - e. On the management page for DS5000, select a device in the **Device Serial Number** column.
 - f. click **Modify Device**.
 - g. Alternatively, right-click a drive and then select **Modify Device**, or double-click a device entry.
 - Command-line interface:

In the *WAS_HOME/bin* directory, start a **wsadmin** session by using Jython. Log on to **wsadmin** with an authorized user ID, such as the SKLMAdmin user ID. For example, on Windows systems, navigate to the *drive:\Program Files (x86)\IBM\WebSphere\AppServer\bin* directory and type:

 - Windows systems:

```
wsadmin -username SKLMAdmin -password mypwd -lang jython
```
 - Systems such as AIX or Linux:

```
./wsadmin.sh -username SKLMAdmin -password mypwd -lang jython
```
2. Modify a device:
 - Graphical user interface:

On the Modify Device dialog, type the changed information. Then, click **Modify Device**.

- Command-line interface:

Type `tklmDeviceUpdate` to update a device. You must specify the device uuid and the attributes that change. For example, type:

```
print AdminTask.tklmDeviceUpdate
  ('[-uuid DEVICE-15d499ad-3ad8-3333-8c84-64cb9e11d990
    -attributes "{description myDevice}"]')
```

- REST interface:

Use **Device Update REST Service** to update a device. For example, send the following HTTP request:

```
PUT https://localhost:9080/SKLM/rest/v1/devices
Content-Type: application/json
Accept : application/json
Authorization: SKLMAuth authId=139aeh34567m
{"uuid":"DEVICE-64c588ad-5ed8-4934-8c84-64cb9e11d990","attributes":
 "description myDevice"}
```

3. A success indicator varies, depending on the interface:

- Graphical user interface:
The device information is changed in the table.
- Command-line interface:
A completion message indicates success.
- Rest interface:
The status code 200 OK indicates success.

What to do next

Next, you might verify that the changes are made.

Deleting a device

You might delete a device such as a DS5000 storage server. Deleting the device removes the device serial number and its key data from the IBM Security Key Lifecycle Manager database.

About this task

If the device in the DS5000 device family and machine affinity is enabled, deleting the device also deletes any relationship between a device and a machine.

You can use the Delete menu item, the **tklmDeviceDelete** command, or **Device Delete REST Service** to delete a device. Your role must have a permission to the delete action and a permission to the appropriate device group.

Procedure

1. Navigate to the appropriate page or directory:
 - Graphical user interface:
 - a. Log on to the graphical user interface.
 - b. In the Key and Device Management section on Welcome page, select **DS5000**.
 - c. Click **Go to > Manage keys and devices**.
 - d. Alternatively, right-click **DS5000** and select **Manage keys and devices**.
 - e. On the management page for DS5000, select a device.
 - f. click **Delete**.
 - g. Alternatively, right-click a drive and then select **Delete**.

- Command-line interface:

In the `WAS_HOME/bin` directory, start a **wsadmin** session by using Jython. Log on to **wsadmin** with an authorized user ID, such as the SKLMAdmin user ID. For example, on Windows systems, navigate to the `drive:\Program Files (x86)\IBM\WebSphere\AppServer\bin` directory and type:

- Windows systems:

```
wsadmin -username SKLMAdmin -password mypwd -lang jython
```

- Systems such as AIX or Linux:

```
./wsadmin.sh -username SKLMAdmin -password mypwd -lang jython
```

2. Using the command-line interface, run the **tklmMachineDeviceList** command or use **Machine Device List REST Service** to obtain the uuid of the device that you intend to delete. Use the **tklmMachineDeviceDelete** command or **Machine Device Delete REST Service** to delete any associations that the device has with machines.

For example, type:

```
print AdminTask.tklmMachineDeviceList
(['-machineID 304238303030343700000000000000'])

print AdminTask.tklmMachineDeviceDelete
(['-deviceUUID DEVICE-663b6d37-e6d5-4c9f-af90-e40e48d27f3c
-machineID 304238303030343700000000000000'])
```

You can send the following HTTP requests:

```
GET https://localhost:9080/SKLM/rest/v1/machines/device?machineID=
304238303030343700000000000000
Content-Type: application/json
Accept : application/json
Authorization: SKLMAuth authId=139aeh34567m

DELETE https://localhost:9080/SKLM/rest/v1/machines/device
Content-Type: application/json
Accept : application/json
Authorization: SKLMAuth authId=139aeh34567m
{"deviceUUID":"DEVICE-663b6d37-e6d5-4c9f-af90-e40e48d27f3c","machineID":
"304238303030343700000000000000"}
```

3. Delete the device:

- Graphical user interface:

On the Confirm dialog, read the confirmation message before you delete the device. Deleting the device removes the device serial number and its key data from the IBM Security Key Lifecycle Manager database.

Then, click **OK**.

- Command-line interface:

Type **tklmDeviceDelete** to delete a device. You must specify the uuid. For example, type:

```
print AdminTask.tklmDeviceDelete
(['-uuid DEVICE-74386920-148c-47b2-a1e2-d19194b315cf'])
```

- REST interface:

Use **Device Delete REST Service** to delete a device. For example, you can send the following HTTP request:

```
DELETE https://localhost:9080/SKLM/rest/v1/devices/DEVICE-74386920-148c-
47b2-a1e2-d19194b315cf
Content-Type: application/json
Accept : application/json
Authorization: SKLMAuth authId=139aeh34567m
```

4. A success indicator varies, depending on the interface:

- Graphical user interface:

The device is removed from the table.

- Command-line interface:
A completion message indicates success.
- Rest interface:
The status code 200 OK indicates success.

Adding keys

You might add more keys for use with DS5000 storage servers.

About this task

You can use the Add Key dialog, the **tklmSecretKeyCreate** command, or **Secret Key Create REST Service** to create one or more symmetric keys in the existing group. Your role must have a permission to the create action and a permission to the appropriate device group.

Before you begin, determine your site policy for naming key prefixes.

Procedure

1. Navigate to the appropriate page or directory:

- Graphical user interface:
 - a. Log on to the graphical user interface.
 - b. In the Key and Device Management section on Welcome page, select **DS5000**.
 - c. Click **Go to > Manage keys and devices**.
 - d. Alternatively, right-click **DS5000** and select **Manage keys and devices**.
 - e. On the management page for DS5000, click **Add**.
 - f. Click **More Keys**.

- Command-line interface:

In the *WAS_HOME/bin* directory, start a **wsadmin** session by using Jython. Log on to **wsadmin** with an authorized user ID, such as the SKLMAdmin user ID. For example, on Windows systems, navigate to the *drive:\Program Files (x86)\IBM\WebSphere\AppServer\bin* directory and type:

– Windows systems:

```
wsadmin -username SKLMAdmin -password mypwd -lang jython
```

– Systems such as AIX or Linux:

```
./wsadmin.sh -username SKLMAdmin -password mypwd -lang jython
```

- REST interface:
 - Open a REST client.

2. Create keys:

- Graphical user interface

On the Add Key dialog, specify values for the required parameters. Then, click **Add More Keys**.

- Command-line interface:

- a. Use the **tklmGroupList** command obtain the value of the uuid for the key group. For example, type:

```
print AdminTask.tklmGroupList ('[-type keygroup -v y]')
```

The output might look like this example:

```

group name = DS5K-ds5kdevice
group type = KEY
group uuid = KEYGROUP-9c97d9aa-b5f0-41a1-b65f-119756168211
initialization date = 6/4/10 12:00:00 AM GMT-12:00
activation date = 6/4/10 12:00:00 AM GMT-12:00
key[0]:
  uuid: KEY-66b0a3a2-3c52-4088-8772-0a1ddebf5803
  aliases: dsk000000000000000000
  keystore names: defaultKeyStore
key[1]:
  uuid: KEY-3f1230fd-59ef-4c15-82e6-40d68ac5f2ab
  aliases: dsk0000000000000000001
  keystore names: defaultKeyStore
.
. (Remaining elements not shown in this example.)
.

```

- b. Create more keys and store them in the group. For example, type:

```

print AdminTask.tklmSecretKeyCreate ('[-alias abc
-keyStoreName defaultKeyStore -numOfKeys 10 -usage DS5000
-keyGroupUuid KEYGROUP-9c97d9aa-b5f0-41a1-b65f-119756168211]')

```

- REST interface:

- Obtain a unique user authentication identifier to access IBM Security Key Lifecycle Manager REST services. For more information about the authentication process, see Authentication process for REST services.
- To invoke **Group List REST Service**, send the HTTP GET request. Pass the user authentication identifier that you obtained in Step a along with the request message as shown in the following example.

```

GET https://localhost:9080/SKLM/rest/v1/keygroups
Content-Type: application/json
Accept : application/json
Authorization: SKLMAuth authId=139aeh34567m
Accept-Language : en

```

The output might look like this example:

```

Status Code : 200 OK
Content-Language: en
[
{
  "group name": "DS5K-ds5kdevice",
  "group type": "KEY",
  "group uuid": "KEYGROUP-9c97d9aa-b5f0-41a1-b65f-119756168211",
  "initialization date": "6/4/10 12:00:00 AM Central Standard Time",
  "activation date": "6/4/10 12:00:00 AM Central Standard Time",
  "keys":
  [
    {
      "uuid": "KEY-66b0a3a2-3c52-4088-8772-0a1ddebf5803",
      "alias(es)": "dsk000000000000000000",
      "key store name(s)": "defaultKeyStore "
    },
    {
      "uuid": "KEY-3f1230fd-59ef-4c15-82e6-40d68ac5f2ab",
      "alias(es)": "dsk0000000000000000001",
      "key store name(s)": "defaultKeyStore "
    }
  ]
}
.
.
.

```

- Use **Secret Key Create REST Service** to create more keys and store them in the group. For example, you can send the following HTTP request:

```

POST https://localhost:9080/SKLM/rest/v1/keys
Content-Type: application/json
Accept : application/json

```



```
Authorization: SKLMAuth authId=139aeh34567m
{"alias":"abc","numOfKeys":"10","keyGroupUuid":"KEYGROUP-9c97d9aa-
b5f0-41a1-b65f-119756168211","usage":"DS5000"}
```

3. A success indicator varies, depending on the interface:

- Graphical user interface:

The additional keys are visible in the table of keys on the Modify Keys page. Back up new keys before the keys are served to devices.

- Command-line interface:

Completion messages indicate success. Additionally, run the **tklmGroupList** command again to verify that the keys that you added now exist in the key group. For example, type:

```
print AdminTask.tklmGroupList ('[-type keygroup -v y]')
```

- Rest interface:

The status code 200 OK indicates success.

What to do next

Next, you might associate the device with a machine.

Modifying (deleting) keys

You might modify the number of keys that a DS5000 storage server uses by deleting one or more keys.

About this task

Delete keys only when the data protected by those keys is no longer needed. Deleting keys is like erasing the data. After keys are deleted, data that is protected by those keys is not retrievable.

You can use the Modify Keys dialog, the **tklmKeyDelete** command, or **Delete Key REST Service**. Your role must have a permission to the modify action and a permission to the appropriate device group.

Before you begin, determine the changed information, such as the number of keys that you want to delete.

Procedure

1. Navigate to the appropriate page or directory:

- Graphical user interface:
 - a. Log on to the graphical user interface.
 - b. In the Key and Device Management section on Welcome page, select **DS5000**.
 - c. Click **Go to > Manage keys and devices**.
 - d. Alternatively, right-click **DS5000** and select **Manage keys and devices**.

- Command-line interface:

In the *WAS_HOME/bin* directory, start a **wsadmin** session by using Jython. Log on to **wsadmin** with an authorized user ID, such as the SKLMAdmin user ID. For example, on Windows systems, navigate to the *drive:\Program Files (x86)\IBM\WebSphere\AppServer\bin* directory and type:

- Windows systems:

```
wsadmin -username SKLMAdmin -password mypwd -lang jython
```

- Systems such as AIX or Linux:

2. Modify the key information:

- Graphical user interface:

On the management page for DS5000, select a device in the **Device Serial Number** column. Then, click **Modify** > **Keys**. Alternatively, right-click a device and then select **Modify Keys**.

Then, on the Modify Keys dialog, select a key and click **Delete**. Read the confirmation message. Then, click **OK**.

- Command-line interface:

You might delete a key. Your role must have a permission to the delete action and a permission to the appropriate device group. For example, type:

```
print AdminTask.tklmKeyDelete ('[-alias aaa0000000000000000000  
-keyStoreName defaultKeyStore]')
```

- REST interface:

Use **Delete Key REST Service** to delete a key. For example, you can send the following HTTP request:

```
DELETE https://localhost:9080/SKLM/rest/v1/keys/aaa0000000000000000000000000000000
Content-Type: application/json
Accept : application/json
Authorization: SKLMAuth authId=139aeh34567m
```

3. A success indicator varies, depending on the interface:

- Graphical user interface:

A column displays changed data. For example, a deleted key is removed from the table of keys in the Modify Keys dialog.

- Command-line interface:

A completion message indicates success.

- Rest interface:

The status code 200 OK indicates success.

What to do next

Next, you might associate the device with a machine.

IBM Spectrum Scale file management

You can use IBM Spectrum Scale file system to manage keys in IBM Security Key Lifecycle Manager.

IBM Spectrum Scale (formerly GPFS) is a high performance shared-disk file management solution that provides fast, reliable access to data from multiple nodes in a cluster environment. Applications can readily access files using standard file system interfaces, and the same file can be accessed concurrently from multiple nodes.

IBM Spectrum Scale provides support for file encryption that ensures both secure storage and secure deletion of data. IBM Spectrum Scale manages encryption through the use of encryption keys and encryption policies.

See IBM Spectrum Scale documentation for more information http://www-01.ibm.com/support/knowledgecenter/SSFKN_4.1.0/gpfs.v4r1_welcome.html.

Administering certificates and keys

To administer certificates and keys, you might want to add, modify, or delete their associated node names. You can also add keys and a name that is associated with the keys.

About this task

Your role must have a permission to the view action and a permission to the appropriate device group. Use the management page for IBM Spectrum Scale to add, modify, or delete a certificate or key.

Before you begin, examine the columns on the page, which provides buttons to add, modify, or delete a table item.

The table is organized in these information areas:


- In left columns, information about certificates indicates the certificate UUID, certificate name, and the endpoint count. The endpoint count is the number of endpoints that are using this certificate.
- In right columns, information about keys indicates the key UUID and the key name that the certificates on the left have access to.

Procedure

1. Log on to the graphical user interface.
 - a. In the Key and Device Management section on Welcome page, select **IBM Spectrum Scale**.
 - b. Click **Go to > Manage keys and devices**.
 - c. Alternatively, right-click **IBM Spectrum Scale** and select **Manage keys and devices**.
2. You can add, modify, or delete a key or certificate.

You might do these administrative tasks:

- Refresh the list.

Click the refresh icon  to refresh items in the table.

- Add

Click **Add**.

- Certificate

On the Add Certificate dialog, type a name and the file name and location of a certificate. Then, click **Add**.

- Key

On the Add Key dialog, specify the information according to your requirements, such as the number of keys to create, up to a maximum of 100 keys. Then, click **Add**.

A success indicator varies, showing the addition of a certificate or keys.

- Modify

To change a certificate or keys, select the certificate or key and then click **Modify**. Alternatively, right-click the selected certificate or key. Then, click **Modify**.

- Certificate

Specify changes on the Modify Certificate dialog. Then, click **Modify**. Your role must have a permission to the modify action and a permission to the appropriate device group.

- Key

Specify changes on the Modify Key dialog. Then, click **Modify**. Your role must have a permission to the delete action and a permission to the appropriate device group.

A success indicator varies, showing a change in a column for the certificate or key.

- Delete

To delete a certificate or key, select the certificate or key, and then click **Delete**. Alternatively, right-click the selected certificate or key, and then click **Delete**.

Metadata for the certificate that you delete is removed from the IBM Security Key Lifecycle Manager database. Key data is also removed. To confirm deletion, click **OK**. Your role must have a permission to the delete action and a permission to the appropriate device group.

A success indicator is deletion of the certificate from the table.

Adding a certificate

You might add more certificates for use with IBM Security Key Lifecycle Manager.

About this task

You can use the Add Certificate dialog to add a certificate. Your role must have a permission to the create action and a permission to the appropriate device group.

Procedure

1. Log on to the graphical user interface.
2. In the Key and Device Management section on Welcome page, select **IBM Spectrum Scale**.
3. Click **Go to > Manage keys and devices**.
4. Alternatively, right-click **IBM Spectrum Scale** and select **Manage keys and devices**.
5. On the management page for IBM Spectrum Scale, click **Add**.
6. Click **Certificate**.
7. On the Add Certificate dialog, specify the information according to the requirements.
8. Click **Add**.

The certificate is added to the table.

Modifying a certificate

You might modify information about a certificate in the IBM Security Key Lifecycle Manager database.

About this task

You can use the Modify Certificate dialog to update a certificate. Your role must have a permission to the modify action and a permission to the appropriate device group.

Procedure

1. Log on to the graphical user interface.
2. In the Key and Device Management section on Welcome page, select **IBM Spectrum Scale**.

3. Click **Go to > Manage keys and devices**.
4. Alternatively, right-click **IBM Spectrum Scale** and select **Manage keys and devices**.
5. On the management page for IBM Spectrum Scale, select a certificate.
6. Click **Modify**.
7. Alternatively, right-click a certificate and then select **Modify**, or double-click a certificate entry.
8. On the Modify Certificate dialog, type the changed information.
9. Click **Modify**.

The certificate information is changed in the table.

What to do next

Next, you might verify that the changes are made.

Deleting a certificate

You might delete a selected certificate, which can be in any state, such as active.

About this task

You can use the Delete menu item to delete a certificate. Your role must have a permission to the delete action and a permission to the appropriate device group.

Procedure

1. Log on to the graphical user interface.
2. In the Key and Device Management section on Welcome page, select **IBM Spectrum Scale**.
3. Click **Go to > Manage keys and devices**.
4. Alternatively, right-click **IBM Spectrum Scale** and select **Manage keys and devices**.
5. On the management page for IBM Spectrum Scale, select a certificate.
6. Click **Delete**.
7. Alternatively, right-click a certificate and then select **Delete**.
8. On the Confirm dialog, read the confirmation message to verify that the correct certificate was selected before you delete the certificate. Then, click **OK**.

The certificate is removed from the table.

Adding keys

You might add keys for use with IBM Spectrum Scale.

About this task

You can use the Add Key dialog to create one or more keys in the existing group. Your role must have a permission to the create action and a permission to the appropriate device group.

Before you begin, determine your site policy for naming key prefixes.

Procedure

1. Log on to the graphical user interface.
2. In the Key and Device Management section on Welcome page, select **IBM Spectrum Scale**.

3. Click **Go to > Manage keys and devices**.
4. Alternatively, right-click **IBM Spectrum Scale** and select **Manage keys and devices**.
5. On the management page for IBM Spectrum Scale, click **Add**.
6. Click **Key**.
7. On the Add Key dialog, specify values for the parameters.
8. Click **Add**. The keys that you added are visible in the table of keys. Back up the keys before the keys are served to devices.

Modifying a key

You might modify information about a key in the IBM Security Key Lifecycle Manager database.

About this task

You can use the Modify Key dialog to modify information about a key. Your role must have a permission to the modify action and a permission to the appropriate device group.

Procedure

1. Log on to the graphical user interface.
2. In the Key and Device Management section on Welcome page, select **IBM Spectrum Scale**.
3. Click **Go to > Manage keys and devices**.
4. Alternatively, right-click **IBM Spectrum Scale** and select **Manage keys and devices**.
5. On the management page for IBM Spectrum Scale, select a key.
6. Click **Modify**.
7. Alternatively, right-click a key and then select **Modify**, or double-click a key entry.
8. On the Modify Key dialog, type the changed information. Then, click **Modify**. The key information is changed in the table.

Deleting a key

You might delete a key entry from the IBM Security Key Lifecycle Manager database.

About this task

You can use the Delete menu item to delete a key. Your role must have a permission to the delete action and a permission to the appropriate device group.

Procedure

1. Log on to the graphical user interface.
2. In the Key and Device Management section on Welcome page, select **IBM Spectrum Scale**.
3. Click **Go to > Manage keys and devices**.
4. Alternatively, right-click **IBM Spectrum Scale** and select **Manage keys and devices**.
5. On the management page for IBM Spectrum Scale, select a key.
6. Click **Delete**.

7. Alternatively, right-click a key and then select **Delete**.
8. On the Confirm dialog, read the confirmation message to verify that the correct key was selected before you delete the key. Then, click **OK**. The key information is removed from the table.

Backup and restore

IBM Security Key Lifecycle Manager provides a set of operations to back up and restore current, active files and data.

IBM Security Key Lifecycle Manager creates cross-platform backup files in a manner that is independent of operating systems and directory structure of the server. You can restore the backup files to an operating system that is different from the one it was backed up from. For example, you can restore a backup file that is taken on a Linux system and restore it on a Windows system.

You can use the cross-platform backup utility to run backup operation on earlier versions of IBM Security Key Lifecycle Manager to back up critical data. You can restore these backup files on current version of IBM Security Key Lifecycle Manager across operating systems.

Note: Starting with the IBM Security Key Lifecycle Manager version 2.6 release, the Solaris operating system is not supported. If you are using IBM Security Key Lifecycle Manager on Solaris systems, use the cross-platform backup utility to back up the data. You can then run the restore operation to restore data on a IBM Security Key Lifecycle Manager version 2.6 system that is deployed on any of the supported operating systems, such as Windows, Linux, or AIX.

Backed up files include the following data:

- Data in the IBM Security Key Lifecycle Manager database tables
- Truststore and keystore with the master key
- IBM Security Key Lifecycle Manager configuration files

Your role must have permissions to back up or to restore files.

Failure to back up your critical data properly might result in unrecoverable loss of all access to your encrypted data. Do not encrypt your backup file, or store a backup file on an encrypting device. Failure to back up data might also result in a later inconsistency of the key manager and potential data loss on the storage device.

The IBM Security Key Lifecycle Manager backup and restore operations support the use of AES 256-bit key length for data encryption/decryption to conform to the PCI DSS (Payment Card Industry Data Security Standard) standards for increased data security.

Backup and restore runtime requirements

Backing up and restoring data from backup files for IBM Security Key Lifecycle Manager have several runtime requirements.

Prevent timeout failure by increasing the time interval that is allowed for backup and restore transactions for large key populations. Specify a larger value for the **totalTranLifetimeTimeout** setting in this file:

`WAS_HOME/profiles/KLMProfile/config/cells/
SKLMCell/nodes/SKLMNode/servers/server1/server.xml`

Additionally, these conditions must be true:

- Ensure that the task occurs during a time interval that allows a halt to key serving activity.
- For a backup task, the IBM Security Key Lifecycle Manager server must be running in a normal operational state. The IBM Security Key Lifecycle Manager database instance must be available.
- For a restore task, the IBM Security Key Lifecycle Manager database instance must be accessible through the IBM Security Key Lifecycle Manager data source. Before you start a restore task, ensure that you have the password that was used when the backup file was created. Restored files must be written to the same IBM Security Key Lifecycle Manager server from which the data was previously backed up. Alternatively, the restored files must be written to a replica computer.
- Ensure that the directories, which are associated with the `tklm.backup.dir` property exist. Also, ensure read and write access to these directories for the system and IBM Security Key Lifecycle Manager administrator accounts under which the IBM Security Key Lifecycle Manager server and the DB2 server run.

Backing up critical files

Use the graphical user interface, command-line interface, or REST interface to back up critical files for IBM Security Key Lifecycle Manager.

About this task

You can use the Backup and Restore page. Alternatively, you can use the `tklmBackupRun` command or **Backup Run REST Service** to back up critical data. Your role must have a permission to back up files.

IBM Security Key Lifecycle Manager creates backup files in a manner that is independent of operating systems and directory structure of the server. You can restore the backup files to an operating system that is different from the one it was backed up from.

Note: Backup success messages are system wide. Two administrators might run backup tasks that overlap in time. During this interval, the administrator who starts a second task that fails might see a false success message from the first backup task.

Procedure

1. Navigate to the appropriate page or directory:

- Graphical user interface:
 - a. Log on to the graphical user interface.
 - b. On the Welcome page, click **Backup and Restore**.
- Command-line interface:

In the `WAS_HOME/bin` directory, start a **wsadmin** session by using Jython. Log on to **wsadmin** with an authorized user ID, such as the SKLMAdmin user ID. For example, on Windows systems, navigate to the `drive:\Program Files (x86)\IBM\WebSphere\AppServer\bin` directory and type:

- Windows systems:

```
wsadmin -username SKLMAdmin -password mypwd -lang jython
```


- Systems such as AIX or Linux:
 - `./wsadmin.sh -username SKLMAdmin -password mypwd -lang jython`
 - REST interface:
 - Open a REST client.
2. Create a backup file. Only one backup or restore task can run at a time.
 - Graphical user interface:
 - a. On the **Backup and Restore** table, click **Browse** and specify a backup repository location such as `C:\Program Files (x86)\IBM\WebSphere\AppServer\products\sklm\restore\backup\`
 - b. Click **Create Backup**.
 - c. On the Create Backup page, specify required information such as a value for the encryption password. Ensure that you retain the encryption password for future use in case you restore the backup.
 - d. Click **Create Backup**.
 - Command-line interface:

Type `tklmBackupRun` and specify the required values to create a backup file. For example, type:

```
print AdminTask.tklmBackupRun
('[-backupDirectory C:\\sklmbakup1 -password myBackupPwd]')
```
 - REST interface:
 - a. Obtain a unique user authentication identifier to access IBM Security Key Lifecycle Manager REST services. For more information about the authentication process, see Authentication process for REST services.
 - b. To invoke **Backup Run REST Service**, send the HTTP POST request. Pass the user authentication identifier that you obtained in Step a along with the request message as shown in the following example.


```
POST https://localhost:9080/SKLM/rest/v1/ckms/backups
Content-Type: application/json
Accept : application/json
Authorization: SKLMAuth authId=139aeh34567m
Accept-Language : en
{"backupDirectory":"/sklmbakup1","password":"myBackupPwd"}
```
 3. A message indicates that the backup file was created, or that the backup operation succeeded.

The time stamp on a backup file has a Greenwich Mean Time (GMT) offset represented in RFC 822 format. The file name contains a *+hhmm* or *-hhmm* element to specify a timezone ahead of or behind GMT. For example, a file name might be `sklm_v2.6.0.0_20100123144220-0800_backup.jar`, where `-0800` indicates that the timezone is eight hours behind GMT.

What to do next

Retain the encryption password for future use in case you restore the backup. Review the directory that contains the backup files to ensure that the backup file exists. Do not edit a file in the backup JAR file. The file that you attempt to edit becomes unreadable.

Restoring a backup file

Use the graphical user interface, command-line interface, or REST interface to restore a backup file for IBM Security Key Lifecycle Manager.

About this task

You can use the Backup and Restore page to restore a backup file. Alternatively, you can use the **tklmBackupRunRestore** command or **Backup Run Restore REST Service** to restore the file. Your role must have a permission to restore files.. IBM Security Key Lifecycle Manager creates backup files in a manner that is independent of operating systems and directory structure of the application. You can restore the backup files to an operating system that is different from the one it was backed up from.

Before you start a restore task, isolate the system for maintenance. Take a backup of the existing system. You can later use this backup to bring the system back to original state if any issues occur during the restore process. You must restart the IBM Security Key Lifecycle Manager server immediately after the restore occurs. Verify the environment before you bring the IBM Security Key Lifecycle Manager server back into production.

Procedure

1. Navigate to the appropriate page or directory:
 - Graphical user interface:
 - a. Log on to the graphical user interface.
 - b. On the Welcome page, click **Backup and Restore**.
 - Command-line interface:

In the *WAS_HOME/bin* directory, start a **wsadmin** session by using Jython. Log on to **wsadmin** with an authorized user ID, such as the SKLMAdmin user ID. For example, on Windows systems, navigate to the *drive:\Program Files (x86)\IBM\WebSphere\AppServer\bin* directory and type:

 - Windows systems:

```
wsadmin -username SKLMAdmin -password mypwd -lang jython
```
 - Systems such as AIX or Linux:

```
./wsadmin.sh -username SKLMAdmin -password mypwd -lang jython
```
 - REST interface:
 - Open a REST client.
2. Restore a selected backup file. Only one backup or restore task can run at a time. If you restore a file to a replica computer, copy the file to that computer by using media such as a disk, or electronic transmission.
 - Graphical user interface:
 - a. Click **Display Backups** to display the backup files that you want to restore.
 - b. On the **Backup and Restore** table, select a backup file that is listed in the table.
 - c. Click **Restore from Backup**.
 - **Note:**
 - If you applied a fix pack on distributed systems, do not attempt to restore the backup files that were created before the fix pack application.
 - d. On the Restore Backup page, specify the encryption password that was used to create the backup file.
 - e. Click **Restore Backup**.
 - Command-line interface:

Type `tklmBackupRunRestore` and specify the required information such as the path and backup file name. Specify the encryption password that was used to create the backup file. For example, type:

```
print AdminTask.tklmBackupRunRestore
('[-backupFilePath /opt/mysklmbackups/sklm_v2.6.0.0_20150705235417-1200_backup
-password myBackupPwd]')
```

- REST interface:
 - a. Obtain a unique user authentication identifier to access IBM Security Key Lifecycle Manager REST services. For more information about the authentication process, see [Authentication process for REST services](#).
 - b. To invoke **Backup Run Restore REST Service**, send the HTTP POST request. Pass the user authentication identifier that you obtained in Step a along with the request message as shown in the following example.

```
POST https://localhost:9080/SKLM/rest/v1/ckms/restore
Content-Type: application/json
Accept : application/json
Authorization: SKLMAuth authId=139aeh34567m
Accept-Language : en
{"backupFilePath":"/opt/mysklmbackups/sklm_v2.5.0.0_20130705235417-1200_
backup.jar","password":"myBackupPwd"}
```

3. A message indicates that the restore operation succeeded.

Results

The IBM Security Key Lifecycle Manager server automatically restarts after a backup file is restored when the **autoRestartAfterRestore** property value is true (default value) in the `SKLMConfig.properties` file.

Note: After automatic restart of the IBM Security Key Lifecycle Manager server, the windows WebSphere Application Server service status is not refreshed and is shown as stopped.

What to do next

Then, determine whether the server is at the expected state. For example, you might examine the keystore to see whether a certificate that had problems before the backup file restore is now available for use.

Installing Java Cryptography Extension unlimited strength jurisdiction policy files

You must install Java Cryptography Extension (JCE) unlimited strength jurisdiction policy files if the IBM Security Key Lifecycle Manager backup operation uses AES 256-bit key for data encryption.

About this task

Note: In the current version, after you install IBM Security Key Lifecycle Manager, the AES 256-bit master key is generated by default and the JCE unlimited strength jurisdiction policy files are installed in the server.

Procedure

1. Go to the `ibm.com` website at <https://www14.software.ibm.com/webapp/iwm/web/preLogin.do?source=jcesdk..>
2. Specify your `ibm.com` website user ID and the password.

3. Download the `unrestrictedpolicyfiles.zip` file.
4. Stop the WebSphere Application Server.
5. Extract the contents of the compressed file to the `<AppServer>\java\jre\lib\security` directory. For example:

Windows

```
C:\Program Files (x86)\IBM\WebSphere\AppServer\java\jre\lib\security
```

Linux `/opt/IBM/WebSphere/AppServer/java/jre/lib/security`

6. Restart WebSphere Application Server.

Starting and stopping the IBM Security Key Lifecycle Manager server on distributed systems

You might want to use the **startServer** or **stopServer** command to start or stop the IBM Security Key Lifecycle Manager server. For example, after a restore task completes, restart the IBM Security Key Lifecycle Manager server.

About this task

The IBM Security Key Lifecycle Manager server automatically restarts after a backup file is restored when the **autoRestartAfterRestore** property value is true (default value) in the `SKLMConfig.properties` file.

Scripts to start and stop the IBM Security Key Lifecycle Manager server are in the `WAS_HOME/bin` directory.

Procedure

1. Navigate to the `WAS_HOME/bin` directory.
2. Start or stop the server.

- Start

On Windows systems:

```
startServer.bat server1
```

On systems such as Linux or AIX:

```
./startServer.sh server1
```

- Stop

On Windows systems:

```
stopServer.bat server1
```

On systems such as Linux or AIX:

```
./stopServer.sh server1
```

Global security is enabled by default. Enter the user ID and password of the WebSphere Application Server administrator as parameters to the `stopServer` script. The script prompts for these parameters when they are omitted, but you can specify them on the command line:

On Windows systems:

```
stopServer.bat server1 -username wasadmin -password mypwd
```

On systems such as Linux or AIX:

```
./stopServer.sh server1 -username wasadmin -password mypwd
```

What to do next

Determine whether IBM Security Key Lifecycle Manager is running. For example, open IBM Security Key Lifecycle Manager in a web browser and log in.

Enabling global security

Conditions might occur in which you must enable global security.

About this task

Do not disable global security when you use IBM Security Key Lifecycle Manager.

Procedure

1. To enable global security, log in as the WebSphere Application Server administrator WASAdmin.
2. In the navigation bar, click **Security**.
3. Click **Secure administration, applications and infrastructure**.
4. Check the **Enable administrative security** check box.
Ensure that **Enable application security** is also selected and that **Use Java 2 security to restrict application access to local resources** is *not* selected.
5. Click **Apply**.
6. Click **Save** in the Messages box. Click **Logout**.
7. Stop and restart the server.
8. Reload the IBM Security Key Lifecycle Manager login page. Verify that the page requires a password.

Disabling global security

Conditions might occur in which you must disable global security.

About this task

Do not disable global security when you use IBM Security Key Lifecycle Manager.

Procedure

1. To disable global security, log in as the WebSphere Application Server administrator WASAdmin.
2. In the navigation bar, click **Security**.
3. Click **Secure administration, applications and infrastructure**.
4. Clear the **Enable administrative security** check box.
5. Click **Apply**.
6. Click **Save** in the Messages box. Click **Logout**.
7. Stop and restart the server.
8. Reload the IBM Security Key Lifecycle Manager login page. Verify that the page does *not* require a password.

Deleting a backup file

Use the graphical user interface or command-line interface to delete a backup file for IBM Security Key Lifecycle Manager. For example, you might delete a backup file for which a business needs no longer exists.

About this task

You can use the Backup and Restore page to delete a backup file.

Your role must have a permission to back up files.

Procedure

1. Log on to the graphical user interface.
2. On the Welcome page, click **Backup and Restore**.
3. On the **Backup and Restore** table, select a backup file that is listed in the table.
4. Click **Delete Backup** and confirm that you want to delete the file.

What to do next

Examine the directory in which the backup files are stored to determine whether the specified file was deleted.

Running backup and restore tasks on the command-line or REST interface

You might use the command-line interface or REST interface for more backup and restore tasks that are not available on the graphical user interface.

About this task

Before you begin, obtain the password. Your role must have permissions to back up or to restore files.

Procedure

1. Navigate to the appropriate directory and log on:
 - Command-line interface:

In the *WAS_HOME/bin* directory, start a **wsadmin** session by using Jython. Log on to **wsadmin** with an authorized user ID, such as the SKLMAdmin user ID. For example, on Windows systems, navigate to the *drive:\Program Files (x86)\IBM\WebSphere\AppServer\bin* directory and type:

 - Windows systems:

```
wsadmin -username SKLMAdmin -password mypwd -lang jython
```
 - Systems such as AIX or Linux:

```
./wsadmin.sh -username SKLMAdmin -password mypwd -lang jython
```
 - REST interface:
 - Open a REST client.
2. Complete the task:
 - Command-line interface:

tklmbBackupGetProgress

Type `tklmbBackupGetProgress` to determine the current phase of a backup task that is running. For example, type:

```
print AdminTask.tklmbBackupGetProgress()
```

tklmbBackupGetRestoreProgress

Type `tklmbBackupGetRestoreProgress` to determine the current phase of a restore task that is running. For example, type:

```
print AdminTask.tklmBackupGetRestoreProgress()
```

tklmBackupGetRestoreResult

Type `tklmBackupGetRestoreResult` to determine the success or failure of a completed restore task. For example, type:

```
print AdminTask.tklmBackupGetRestoreResult()
```

tklmBackupGetResult

Type `tklmBackupGetResult` to determine the success or failure of a completed backup task. For example, type:

```
print AdminTask.tklmBackupGetResult()
```

tklmBackupIsRestoreRunning

Type `tklmBackupIsRestoreRunning` to determine whether the restore task is running. For example, type:

```
print AdminTask.tklmBackupIsRestoreRunning()
```

tklmBackupIsRunning

Type `tklmBackupIsRunning` to determine whether the backup task is running. For example, type:

```
print AdminTask.sklmBackupIsRunning()
```

tklmBackupList

Type `tklmBackupList` to list the backup files in a directory. For example, type:

```
print AdminTask.tklmBackupList  
(['[-backupDirectory C:\\\\tipbak1\\\\tklmbakup1 -v y]'])
```

- REST interface:
 - a. Obtain a unique user authentication identifier to access IBM Security Key Lifecycle Manager REST services. For more information about the authentication process, see Authentication process for REST services.
 - b. To invoke the REST service, send the HTTP request. Pass the user authentication identifier that you obtained in Step a along with the request message as shown in the following example.

Backup Get Progress REST Service

Use **Backup Get Progress REST Service** to determine the current phase of a backup task that is running. For example, you can send the following HTTP request:

```
GET https://localhost:9080/SKLM/rest/v1/ckms/backups/progress  
Content-Type: application/json  
Accept : application/json  
Authorization: SKLMAuth authId=139aeh34567m  
Accept-Language : en
```

Backup Get Restore Progress REST Service

Use **Backup Get Restore Progress REST Service** to determine the current phase of a restore task that is running. For example, you can send the following HTTP request:

```
GET https://localhost:9080/SKLM/rest/v1/ckms/restore/progress  
Content-Type: application/json  
Accept: application/json  
Authorization: SKLMAuth authId=139aeh34567m  
Accept-Language: en
```

Backup Get Restore Result REST Service

Type **Backup Get Restore Result REST Service** to determine the

success or failure of a completed restore task. For example, you can send the following HTTP request:

```
GET https://localhost:9080/SKLM/rest/v1/ckms/restore/result
Content-Type: application/json
Accept : application/json
Authorization: SKLMAuth authId=139aeh34567m
Accept-Language : en
```

Backup Get Result REST Service

Type **Backup Get Result REST Service** to determine the success or failure of a completed backup task. For example, you can send the following HTTP request:

```
GET https://localhost:9080/SKLM/rest/v1/ckms/backups/result
Content-Type: application/json
Accept : application/json
Authorization: SKLMAuth authId=139aeh34567m
Accept-Language : en
```

Backup List REST Service

Use **Backup List REST Service** to list the backup files in a directory. For example, you can send the following HTTP request:

```
GET https://localhost:9080/SKLM/rest/v1/ckms/backups?backupDirectory=
/sklmbackup
Content-Type: application/json
Accept : application/json
Authorization: SKLMAuth authId=139aeh34567m
Accept-Language : en
```

A completion message indicates success.

- Command-line interface:

Completion messages indicate success.

- Rest interface:

The status code 200 OK indicates success.

Backup and restore operations for earlier versions of IBM Security Key Lifecycle Manager

You can use the cross-platform backup utility of current version of IBM Security Key Lifecycle Manager to run backup operation on earlier versions of IBM Security Key Lifecycle Manager to back up critical data. You can restore these backup files on current version of IBM Security Key Lifecycle Manager across operating systems by using the restore utility.

Backing up Encryption Key Manager, Version 2.1 data

Use the IBM Security Key Lifecycle Manager, Version 2.6 backup utility to create Encryption Key Manager, Version 2.1 backup files.

Before you begin

- You must install IBM Security Key Lifecycle Manager, Version 2.6 on a system.
- Ensure that the Encryption Key Manager folder contains the configuration file, keystore files, other data files and folders that are related to drivetable, key groups, and metadata.

About this task

You can use the backup utility to create cross-platform backup files in a manner that is independent of operating systems and directory structure of the server. You can restore these cross-platform compatible backup files on a system with IBM

Security Key Lifecycle Manager, Version 2.6 across operating systems.

Procedure

1. Copy the Encryption Key Manager folder and all other necessary files to a system where IBM Security Key Lifecycle Manager, Version 2.6 is installed.
2. Ensure that the KeyManagerConfig.properties file and the following files that are mentioned in the KeyManagerConfig.properties file are copied.

Note: You must edit the KeyManagerConfig.properties configuration file in Encryption Key Manager folder to specify absolute paths of keystore and other data files as shown in the following example.

```
Admin.ssl.keystore.name=C:/EKM21/test.keys.ssl
Admin.ssl.truststore.name=C:/EKM21/test.keys.ssl
TransportListener.ssl.truststore.name=C:/EKM21/test.keys.ssl
TransportListener.ssl.keystore.name=C:/EKM21/test.keys.ssl
config.keystore.file=C:/EKM21/test.keys.jceks
config.drivetable.file.url=FILE\C:/EKM21/filedrive.table
Audit.handler.file.directory=C:/audit
Audit.metadata.file.name=C:/EKM21/metadata/EKMData.xml
config.keygroup.xml.file=FILE\C:/EKM21/KeyGroups.xml
```

3. Locate backup utilities folder in the system where version 2.6 is installed.

Windows

```
<SKLM_INSTALL_HOME>\migration\utilities\ekm21
```

Default location is C:\Program Files (x86)\IBM\SKLMV26\migration\utilities\ekm21.

Linux <SKLM_INSTALL_HOME>/migration/utilities/ekm21

Default location is /opt/IBM/SKLMV26/migration/utilities/ekm21.

4. Edit backup.properties in the backup utilities folder to configure properties as shown in the following example. You must set values for all the properties, except for the BACKUP_DIR property (optional).

If you do not specify the value for BACKUP_DIR, the backup file is created in the backup subfolder under the same directory from where you run the backup utility.

Note: On Windows operating system, the backup.properties file that you use for backup operations must not contain the property keys and values with leading or trailing spaces.

Windows

```
KLM_VERSION=2.1
BACKUP_DIR=C:\\ekm_backup
EKM_HOME=C:\\EKM21
BACKUP_PASSWORD=passw0rd123
JAVA_HOME=C:\\Program Files (x86)\\IBM\\WebSphere\\AppServer\\java
```

Linux

```
KLM_VERSION=2.1
BACKUP_DIR=/ekm_backup
EKM_HOME=/EKM21
BACKUP_PASSWORD=passw0rd123
JAVA_HOME=/opt/IBM/WebSphere/AppServer/java
```

Note: On Windows system, when you specify path in the properties file, use either “/” or “\\” as path separator as shown in following example.

```
C:\\ekm_backup
```

Or

C:/ekm_backup

5. Open a command prompt and run the backup utility.

Windows

Go to the <SKLM_INSTALL_HOME>\migration\utilities\ekm21 directory and run the following command:

backupEKM21.bat

Linux Go to the <SKLM_INSTALL_HOME>/migration/utilities/ekm21 directory and run the following command:

backupEKM21.sh

What to do next

- Review the directory that contains backup files to ensure that the backup file exists. The backup files are created in the location that you specified for BACKUP_DIR in the backup.properties file.
- Check the backup.log file for errors or exceptions. The backup.log file is created in the same directory where you run the backup utility. For a successful backup operation, ensure that there are no errors or exceptions in the log file.
- Retain the backup password for future use in case you restore the backup.
- Do not edit a file in the backup archive. The file that you attempt to edit becomes unreadable.

Restoring Encryption Key Manager, Version 2.1 backup files

You can restore the Encryption Key Manager, Version 2.1 cross-platform backup files on a system with IBM Security Key Lifecycle Manager, Version 2.6 by using graphical user interface, command-line interface, REST interface, or the migration restore script.

Before you begin

Install IBM Security Key Lifecycle Manager, Version 2.6 on a system. You must have the Encryption Key Manager backup file and ensure that you have the password that you used when the backup file was created.

Note: Your role must have the permissions to run backup and restore operations.

About this task

You can restore Encryption Key Manager cross-platform compatible backup files on a system with IBM Security Key Lifecycle Manager, Version 2.6 across operating systems.

Before you start a restore task, isolate the system for maintenance. Take a backup of the existing system. You can later use this backup to bring the system back to original state if any issues occur during the restore process. You must restart the IBM Security Key Lifecycle Manager server immediately after the restore occurs. Verify the environment before you bring the IBM Security Key Lifecycle Manager server back into production.

Procedure

1. Log on to the system where IBM Security Key Lifecycle Manager, Version 2.6 is installed.

2. Copy the backup file, for example
sklm_vEKM21_20150820113253+0530_backup.jar, from Encryption Key Manager, Version 2.1 system to a directory of your choice.
3. Restore the backup file by using any of the following methods.

Graphical user interface	<ol style="list-style-type: none"> 1. Log on to the graphical user interface as an authorized user, for example, SKLMAdmin. 2. On the Welcome page, click Backup and Restore. 3. In the Backup repository location field, specify full path of the directory that contains Encryption Key Manager backup file. To locate the directory, click Browse. 4. Click Display Backups to display the backup files that you want to restore. 5. In the Backup and Restore table, select a backup file. 6. Click Restore From Backup. 7. On the Restore Backup page, specify the backup password that you used to create the backup file. 8. Click Restore Backup. 9. Restart IBM Security Key Lifecycle Manager server.
Command-line interface	<ol style="list-style-type: none"> 1. Go to the WAS_HOME/bin directory. For example, <ul style="list-style-type: none"> Windows <pre>cd drive:\Program Files (x86)\IBM\WebSphere\ AppServer\bin</pre> Linux <pre>cd /opt/IBM/WebSphere/AppServer/bin</pre> 2. Start the wsadmin interface by using an authorized user ID, such as SKLMAdmin. For example, <ul style="list-style-type: none"> Windows <pre>wsadmin -username SKLMAdmin -password mypwd -lang jython</pre> Linux <pre>./wsadmin.sh -username SKLMAdmin -password mypwd -lang jython</pre> 3. Run the tklmBackupRunRestore CLI command by specifying the parameters such as the backup file name with its full path and backup password that you used to create the backup as shown in the following example. <pre>print AdminTask.tklmBackupRunRestore ('[-backupFilePath /opt/mysklmbackups/sklm_vEKM21_20150820113253+0530_backup.jar -password myBackupPwd]')</pre> 4. Restart IBM Security Key Lifecycle Manager server.

REST interface	<ol style="list-style-type: none"> 1. Open a REST client. 2. Obtain a unique user authentication identifier to access IBM Security Key Lifecycle Manager REST services. For more information about the authentication process, see Authentication process for REST services. 3. To invoke Backup Run Restore REST Service, send the HTTP POST request with backup file name with its full path and backup password as parameters. Pass the user authentication identifier that you obtained in Step b along with the request message as shown in the following example. POST https://localhost:9080/SKLM/rest/v1/ckms/restore Content-Type: application/json Accept: application/json Authorization: SKLMAuth authId=139aeh34567m Accept-Language: en {"backupFilePath":"/opt/mysklmbackups/sklm_vEKM21_20150820113253+0530_backup.backup.jar","password":"myBackupPwd"} 4. Restart IBM Security Key Lifecycle Manager server.
-----------------------	--

Migration restore script	<ol style="list-style-type: none"> 1. Locate the IBM Security Key Lifecycle Manager restore utilities. <ul style="list-style-type: none"> Windows <pre><SKLM_INSTALL_HOME>\migration\utilities\ekm21</pre> <p>Default location is C:\Program Files (x86)\IBM\SKLMV26\migration\utilities\ekm21.</p> Linux <pre><SKLM_INSTALL_HOME>/migration/utilities/ekm21</pre> <p>Default location is /opt/IBM/SKLMV26/migration/utilities/ekm21.</p> 2. Edit restore.properties in the ekm21 folder to configure properties as shown in the following example: Note: On Windows operating system, the restore.properties file that you use for restore operations must not contain the property keys and values with leading or trailing spaces. <ul style="list-style-type: none"> Window <pre>WAS_HOME=C:\\Program Files (x86)\\IBM\\WebSphere\\AppServer BACKUP_PASSWORD=passwd123 DB_PASSWORD=sk1mdb26 RESTORE_FILE=C:\\ekm_restore\\sklm_vEKM21_20150924024117-0400_backup. WAS_USER_PWD=wasadmin RESTORE_USER_ROLES=n</pre> Linux <pre>WAS_HOME=/opt/IBM/WebSphere/AppServer BACKUP_PASSWORD=passwd123 DB_PASSWORD=sk1mdb26 RESTORE_FILE=/ekm_restore/sklm_vEKM21_20150820113253+0530_backup. WAS_USER_PWD=wasadmin RESTORE_USER_ROLES=n</pre> <p>Note: On Windows operating system, when you specify path in the properties file, use either “/” or “\\” as path separator as shown in the following example.</p> <pre>C:\\ekm_restore</pre> <p>Or</p> <pre>C:/ekm_restore</pre> 3. Open a command prompt and run the restore utility. <ul style="list-style-type: none"> Windows <p>Go to the <SKLM_INSTALL_HOME>\migration\utilities\ekm21 directory and run the following command:</p> <pre>restoreEKM21.bat</pre> Linux <p>Go to the <SKLM_INSTALL_HOME>/migration/utilities/ekm21 directory and run the following command:</p> <pre>restoreEKM21.sh</pre> 4. Restart IBM Security Key Lifecycle Manager server.
---------------------------------	--

Backing up IBM Security Key Lifecycle Manager, Version 1.0 (TKLM V1.0) data

Use the IBM Security Key Lifecycle Manager, Version 2.6 backup utility to create IBM Security Key Lifecycle Manager, Version 1.0 (formerly called Tivoli Key Lifecycle Manager, Version 1.0) cross-platform backup files.

Before you begin

You must install IBM Security Key Lifecycle Manager, Version 2.6 on a system. Ensure that the system with IBM Security Key Lifecycle Manager, Version 1.0 with fix pack 7 is available. The keystore must be configured before you run the backup operation.

About this task

You can use the backup utility to create cross-platform backup files in a manner that is independent of operating systems and directory structure of the server. You can restore these cross-platform compatible backup files on a system with IBM Security Key Lifecycle Manager, Version 2.6 across operating systems.

Procedure

Run the followings steps on systems where the IBM Security Key Lifecycle Manager version 2.6 and version 1.0 are installed.

IBM Security Key Lifecycle Manager, Version 2.6	<div><div><div>1. Log on to the system with your user credentials.</div><div>2. Locate the backup utilities folder.</div></div><div><div><div>Windows</div><div><SKLM_INSTALL_HOME>\migration\utilities\v1</div><div>Default location is C:\Program Files (x86)\IBM\SKLMV26\migration\utilities\v1.</div></div><div><div><div>Linux</div><div><SKLM_INSTALL_HOME>/migration/utilities/v1</div><div>Default location is /opt/IBM/SKLMV26/migration/utilities/v1.</div></div></div><div><div>3. Copy the rt.jar and xml.jar files from <WAS_HOME>/java/jre/lib folder to the backup utilities folder that is described in Step b.</div></div></div></div>
--	---

IBM Security Key Lifecycle Manager, Version 1.0	<ol style="list-style-type: none"> 1. Log on to the system with your user credentials. 2. Copy v1 folder from the system where IBM Security Key Lifecycle Manager, Version 2.6 is installed to a local directory of your choice. 3. Edit backup.properties in the v1 folder to configure properties as shown in the following example. You must set values for all the properties, except for the BACKUP_DIR property (optional). If you do not specify the value for BACKUP_DIR, the backup file is created in the backup subfolder under the same directory from where you run the backup utility. Note: On Windows operating system, the backup.properties file that you use for backup operations must not contain the property keys and values with leading or trailing spaces. Windows TKLM_TIP_HOME=C:\\IBM\\tivoli\\tip DB_PASSWORD=tklmb2 KEYSTORE_PASSWORD=Passw0rd TIP_USER_PWD=tipadmin BACKUP_PASSWORD=passw0rd123 BACKUP_DIR=C:\\tklmv1_backup Linux TKLM_TIP_HOME=/opt/IBM/tivoli/tip/ DB_PASSWORD=tklmb2 KEYSTORE_PASSWORD=Passw0rd TIP_USER_PWD=tipadmin BACKUP_PASSWORD=passw0rd123 BACKUP_DIR=/tklmv1_backup Note: On Windows operating system, when you specify path in the properties file, use either “/” or “\\” as path separator as shown in following example. C:\\tklmv1_backup Or C:/tklmv1_backup 4. Open a command prompt and run the backup utility. Windows Go to the v1 directory (see Step b) and run the following command: backupV1.bat Linux Go to the v1 directory (see Step b) and run the following command: backupV1.sh
--	---

What to do next

- Review the directory that contains backup files to ensure that the backup file exists. The backup files are created in the location that you specified for BACKUP_DIR in the backup.properties file.
- Check the backup.log file for errors or exceptions. The backup.log file is created in the same directory where you run the backup utility. For a successful backup operation, ensure that there are no errors or exceptions in the log file.
- Retain the backup password for future use in case you restore the backup.
- Do not edit a file in the backup archive. The file that you attempt to edit becomes unreadable.

Restoring IBM Security Key Lifecycle Manager, Version 1.0 (TKLM V1.0) backup files

You can restore the IBM Security Key Lifecycle Manager, Version 1.0 (formerly called Tivoli Key Lifecycle Manager, Version 1.0) cross-platform backup files on a system with IBM Security Key Lifecycle Manager, Version 2.6 by using the graphical user interface, command-line interface, REST interface, or the migration restore script.

Before you begin

Install IBM Security Key Lifecycle Manager, Version 2.6 on a system. You must have the backup file from the earlier version and ensure that you have the password that you used when the backup file was created.

Note: Your role must have the permissions to run backup and restore operations.

About this task

You can restore IBM Security Key Lifecycle Manager, Version 1.0 cross-platform compatible backup files on a system with IBM Security Key Lifecycle Manager, Version 2.6 across operating systems.

Before you start a restore task, isolate the system for maintenance. Take a backup of the existing system. You can later use this backup to bring the system back to original state if any issues occur during the restore process. You must restart the IBM Security Key Lifecycle Manager server immediately after the restore occurs. Verify the environment before you bring the IBM Security Key Lifecycle Manager server back into production.

Procedure

1. Log on to the system where IBM Security Key Lifecycle Manager, Version 2.6 is installed.
2. Copy the backup file, for example `tklm_v1.0.0.7_20150729013250-0400_migration_backup.jar`, from version 1.0 system to a directory of your choice.
3. Restore the backup file by using any of the following methods.

Graphical user interface	<ol style="list-style-type: none"> 1. Log on to the graphical user interface as an authorized user, for example, SKLMAdmin. 2. On the Welcome page, click Backup and Restore. 3. In the Backup repository location field, specify full path of the directory that contains the version 1.0 backup file. To locate the directory, click Browse. 4. Click Display Backups to display the backup files that you want to restore. 5. In the Backup and Restore table, select a backup file. 6. Click Restore From Backup. 7. On the Restore Backup page, specify the backup password that you used to create the backup file. 8. Click Restore Backup. 9. Restart IBM Security Key Lifecycle Manager server. <p>Note: By using the graphical user interface, you cannot restore roles, users, and groups from IBM Security Key Lifecycle Manager, Version 1.0 backup.</p>
Command-line interface	<ol style="list-style-type: none"> 1. Go to the WAS_HOME/bin directory. For example, <ul style="list-style-type: none"> Windows <pre>cd drive:\Program Files (x86)\IBM\WebSphere\AppServer\bin</pre> Linux <pre>cd /opt/IBM/WebSphere/AppServer/bin</pre> 2. Start the wsadmin interface by using an authorized user ID, such as SKLMAdmin. For example, <ul style="list-style-type: none"> Windows <pre>wsadmin -username SKLMAdmin -password mypwd -lang jython</pre> Linux <pre>./wsadmin.sh -username SKLMAdmin -password mypwd -lang jython</pre> 3. Run the tklmBackupRunRestore CLI command by specifying the parameters such as the backup file name with its full path and backup password that you used to create the backup as shown in the following example. <pre>print AdminTask.tklmBackupRunRestore ('[-backupFilePath /opt/mysklmbackups/tklm_v1.0.0.7_20150729013250-0400_mi -password myBackupPwd]')</pre> 4. Restart IBM Security Key Lifecycle Manager server. <p>Note: By using the command-line interface, you cannot restore roles, users, and groups from IBM Security Key Lifecycle Manager, Version 1.0 backup.</p>

REST interface	<ol style="list-style-type: none"> 1. Open a REST client. 2. Obtain a unique user authentication identifier to access IBM Security Key Lifecycle Manager REST services. For more information about the authentication process, see Authentication process for REST services. 3. To invoke Backup Run Restore REST Service, send the HTTP POST request with backup file name with its full path and backup password as parameters. Pass the user authentication identifier that you obtained in Step b along with the request message as shown in the following example. POST https://localhost:9080/SKLM/rest/v1/ckms/restore Content-Type: application/json Accept: application/json Authorization: SKLMAuth authId=139aeh34567m Accept-Language : en {"backupFilePath":"/opt/mysklmbackups/tklm_v1.0.0.7_20150729013250-0400_migration_backup.jar","password":"myBackupPwd"} 4. Restart IBM Security Key Lifecycle Manager server. <p>Note: By using the REST interface, you cannot restore roles, users, and groups from IBM Security Key Lifecycle Manager, Version 1.0 backup.</p>
----------------	--

Migration restore script	<ol style="list-style-type: none"> 1. Locate the IBM Security Key Lifecycle Manager restore utilities. <p>Windows</p> <pre><SKLM_INSTALL_HOME>\migration\utilities\v1</pre> <p>Default location is C:\Program Files (x86)\IBM\SKLMV26\migration\utilities\v1.</p> <p>Linux <SKLM_INSTALL_HOME>/migration/utilities/v1</p> <p>Default location is /opt/IBM/SKLMV26/migration/utilities/v1.</p> 2. Edit restore.properties in the v1 folder to configure properties as shown in the following example: Note: On Windows operating system, the restore.properties file that you use for restore operations must not contain the property keys and values with leading or trailing spaces. <p>Windows</p> <pre>WAS_HOME=C:\\Program Files (x86)\\IBM\\WebSphere\\AppServer BACKUP_PASSWORD=passwd123 DB_PASSWORD=sklmb26 RESTORE_FILE=C:\\tklmv1_restore\\tklm_v1.0.0.7_20150729013250-0400_ WAS_USER_PWD=wasadmin RESTORE_USER_ROLES=y</pre> <p>Linux</p> <pre>WAS_HOME=/opt/IBM/WebSphere/AppServer BACKUP_PASSWORD=passwd123 DB_PASSWORD=sklmb26 RESTORE_FILE=/tklmv1_restore/tklm_v1.0.0.7_20150729013250-0400_migr WAS_USER_PWD=wasadmin RESTORE_USER_ROLES=y</pre> <p>Note: On Windows operating system, when you specify path in the properties file, use either “/” or “\\” as path separator as shown in the following example.</p> <pre>C:\\tklmv1_restore</pre> <p>Or</p> <pre>C:/tklmv1_restore</pre> 3. Open a command prompt and run the restore utility. <p>Windows</p> <p>Go to the <SKLM_INSTALL_HOME>\migration\utilities\v1 directory and run the following command:</p> <pre>restoreV1.bat</pre> <p>Linux Go to the <SKLM_INSTALL_HOME>/migration/utilities/v1 directory and run the following command:</p> <pre>restoreV1.sh</pre> 4. Restart IBM Security Key Lifecycle Manager server. <p>Note: By using the migration restore script, you can restore roles, users, and groups from IBM Security Key Lifecycle Manager, Version 1.0 backup.</p>
---------------------------------	--

What to do next

Rollovers that are configured for LTO key groups and 3592 certificates are not automatically restored from the earlier versions of IBM Security Key Lifecycle Manager. You must manually set the rollover for certificates and key groups.

For more information, see “Restoring rollover certificates and key groups” on page 152.

Backing up IBM Security Key Lifecycle Manager, Version 2.0 (TKLM V2.0) data

Use the IBM Security Key Lifecycle Manager, Version 2.6 backup utility to create IBM Security Key Lifecycle Manager, Version 2.0 (formerly called Tivoli Key Lifecycle Manager, Version 2.0) backup files.

Before you begin

You must install IBM Security Key Lifecycle Manager, Version 2.6 on a system. Ensure that the system with IBM Security Key Lifecycle Manager, Version 2.0 with fix pack 6 is available. The keystore must be configured before you run the backup operation.

About this task

You can use the backup utility to create cross-platform backup files in a manner that is independent of operating systems and directory structure of the server. You can restore these cross-platform compatible backup files on a system with IBM Security Key Lifecycle Manager, Version 2.6 across operating systems.

Procedure

Run the followings steps on systems where the IBM Security Key Lifecycle Manager version 2.6 and version 2.0 are installed.

IBM Security Key Lifecycle Manager, Version 2.6	<ol style="list-style-type: none">1. Log on to the system with your user credentials.2. Locate the backup utilities folder. Windows <SKLM_INSTALL_HOME>\migration\utilities\v2 Default location is C:\Program Files (x86)\IBM\SKLMV26\migration\utilities\v2. Linux <SKLM_INSTALL_HOME>/migration/utilities/v2 Default location is /opt/IBM/SKLMV26/migration/utilities/v2.3. Copy the rt.jar and xml.jar files from <WAS_HOME>/java/jre/lib folder to the backup utilities folder that is described in Step b.
--	--

IBM Security Key Lifecycle Manager, Version 2.0	<ol style="list-style-type: none"> 1. Log on to the system with your user credentials. 2. Copy v2 folder from the system where IBM Security Key Lifecycle Manager, Version 2.6 is installed to a local directory of your choice. 3. Edit backup.properties in the v2 folder to configure properties as shown in the following example. You must set values for all the properties, except for the BACKUP_DIR property (optional). If you do not specify the value for BACKUP_DIR, the backup file is created in the backup subfolder under the same directory from where you run the backup utility. Note: On Windows operating system, the backup.properties file that you use for backup operations must not contain the property keys and values with leading or trailing spaces. Windows TKLM_TIP_HOME=C:\\IBM\\tivoli\\tiptklmV2 DB_PASSWORD=tklmb2 KEYSTORE_PASSWORD=Passw0rd TIP_USER_PWD=tipadmin BACKUP_PASSWORD=passw0rd123 BACKUP_DIR=C:\\tklmv2_backup Linux TKLM_TIP_HOME=/opt/IBM/tivoli/tiptklmV2/ DB_PASSWORD=tklmb2 KEYSTORE_PASSWORD=Passw0rd TIP_USER_PWD=tipadmin BACKUP_PASSWORD=passw0rd123 BACKUP_DIR=/tklmv2_backup Note: On Windows operating system, when you specify path in the properties file, use either “/” or “\\” as path separator as shown in following example. C:\\tklmv2_backup Or C:/tklmv2_backup 4. Open a command prompt and run the backup utility. Windows Go to the v2 directory (see Step b) and run the following command: backupV2.bat Linux Go to the v2 directory (see Step b) and run the following command: backupV2.sh
--	---

What to do next

- Review the directory that contains backup files to ensure that the backup file exists. The backup files are created in the location that you specified for BACKUP_DIR in the backup.properties file.
- Check the backup.log file for errors or exceptions. The backup.log file is created in the same directory where you run the backup utility. For a successful backup operation, ensure that there are no errors or exceptions in the log file.
- Retain the backup password for future use in case you restore the backup.
- Do not edit a file in the backup archive. The file that you attempt to edit becomes unreadable.

Restoring IBM Security Key Lifecycle Manager, Version 2.0 (TKLM V2.0) backup files

You can restore the IBM Security Key Lifecycle Manager, Version 2.0 (formerly called Tivoli Key Lifecycle Manager, Version 2.0) cross-platform backup files on a system with IBM Security Key Lifecycle Manager, Version 2.6 by using the graphical user interface, command-line interface, REST interface, or the migration restore script.

Before you begin

Install IBM Security Key Lifecycle Manager, Version 2.6 on a system. You must have the backup file from the earlier version and ensure that you have the password that you used when the backup file was created.

Note: Your role must have the permissions to run backup and restore operations.

About this task

You can restore IBM Security Key Lifecycle Manager, Version 2.0 cross-platform compatible backup files on a system with IBM Security Key Lifecycle Manager, Version 2.6 across operating systems.

Before you start a restore task, isolate the system for maintenance. Take a backup of the existing system. You can later use this backup to bring the system back to original state if any issues occur during the restore process. You must restart the IBM Security Key Lifecycle Manager server immediately after the restore occurs. Verify the environment before you bring the IBM Security Key Lifecycle Manager server back into production.

Procedure

1. Log on to the system where IBM Security Key Lifecycle Manager, Version 2.6 is installed.
2. Copy the backup file, for example `tklm_v2.0.0.6_20150729013250-0400_migration_backup.jar`, from version 2.0 system to a directory of your choice.
3. Restore the backup file by using any of the following methods.

Graphical user interface	<ol style="list-style-type: none"> 1. Log on to the graphical user interface as an authorized user, for example, SKLMAdmin. 2. On the Welcome page, click Backup and Restore. 3. In the Backup repository location field, specify full path of the directory that contains the version 2.0 backup file. To locate the directory, click Browse. 4. Click Display Backups to display the backup files that you want to restore. 5. In the Backup and Restore table, select a backup file. 6. Click Restore From Backup. 7. On the Restore Backup page, specify the backup password that you used to create the backup file. 8. Click Restore Backup. 9. Restart IBM Security Key Lifecycle Manager server. <p>Note: By using the graphical user interface, you cannot restore roles, users, and groups from IBM Security Key Lifecycle Manager, Version 2.0 backup.</p>
Command-line interface	<ol style="list-style-type: none"> 1. Go to the WAS_HOME/bin directory. For example, <ul style="list-style-type: none"> Windows <pre>cd drive:\Program Files (x86)\IBM\WebSphere\AppServer\bin</pre> Linux <pre>cd /opt/IBM/WebSphere/AppServer/bin</pre> 2. Start the wsadmin interface by using an authorized user ID, such as SKLMAdmin. For example, <ul style="list-style-type: none"> Windows <pre>wsadmin -username SKLMAdmin -password mypwd -lang jython</pre> Linux <pre>./wsadmin.sh -username SKLMAdmin -password mypwd -lang jython</pre> 3. Run the tklmbBackupRunRestore CLI command by specifying the parameters such as the backup file name with its full path and backup password that you used to create the backup as shown in the following example. <pre>print AdminTask.tklmbBackupRunRestore (['-backupFilePath /opt/my sklmbbackups/tklm_v2.0.0.6_20150729013250-0400_1' '-password myBackupPwd'])</pre> 4. Restart IBM Security Key Lifecycle Manager server. <p>Note: By using the command-line interface, you cannot restore roles, users, and groups from IBM Security Key Lifecycle Manager, Version 2.0 backup.</p>

REST interface	<ol style="list-style-type: none"> 1. Open a REST client. 2. Obtain a unique user authentication identifier to access IBM Security Key Lifecycle Manager REST services. For more information about the authentication process, see Authentication process for REST services. 3. To invoke Backup Run Restore REST Service, send the HTTP POST request with backup file name with its full path and backup password as parameters. Pass the user authentication identifier that you obtained in Step b along with the request message as shown in the following example. POST https://localhost:9080/SKLM/rest/v1/ckms/restore Content-Type: application/json Accept: application/json Authorization: SKLMAuth authId=139aeh34567m Accept-Language: en {"backupFilePath":"/opt/mysklmbackups/tklm_v2.0.0.6_20150729013250-0400_migr backup.jar","password":"myBackupPwd"} 4. Restart IBM Security Key Lifecycle Manager server. <p>Note: By using the REST interface, you cannot restore roles, users, and groups from IBM Security Key Lifecycle Manager, Version 2.0 backup.</p>
----------------	---

Migration restore script	<ol style="list-style-type: none"> 1. Locate the IBM Security Key Lifecycle Manager restore utilities. <ul style="list-style-type: none"> Windows <pre><SKLM_INSTALL_HOME>\migration\utilities\v2</pre> <p>Default location is C:\Program Files (x86)\IBM\SKLMV26\migration\utilities\v2.</p> Linux <pre><SKLM_INSTALL_HOME>/migration/utilities/v2</pre> <p>Default location is /opt/IBM/SKLMV26/migration/utilities/v2.</p> 2. Edit restore.properties in the v2 folder to configure properties as shown in the following example: <p>Note: On Windows operating system, the restore.properties file that you use for restore operations must not contain the property keys and values with leading or trailing spaces.</p> <ul style="list-style-type: none"> Window <pre>WAS_HOME=C:\\Program Files (x86)\\IBM\\WebSphere\\AppServer BACKUP_PASSWORD=passwd123 DB_PASSWORD=sklmb26 RESTORE_FILE=C:\\tklmv2_restore\\tklm_v2.0.0.6_20150729013250-0400_mi WAS_USER_PWD=wasadmin RESTORE_USER_ROLES=y</pre> Linux <pre>WAS_HOME=/opt/IBM/WebSphere/AppServer BACKUP_PASSWORD=passwd123 DB_PASSWORD=sklmb26 RESTORE_FILE=/tklmv2_restore/tklm_v2.0.0.6_20150729013250-0400_mi WAS_USER_PWD=wasadmin RESTORE_USER_ROLES=y</pre> <p>Note: On Windows system, when you specify path in the properties file, use either “/ ” or “\\ ” as path separator as shown in the following example.</p> <pre>C:\\tklmv2_restore</pre> <p>Or</p> <pre>C:/tklmv2_restore</pre> 3. Open a command prompt and run the restore utility. <ul style="list-style-type: none"> Windows <p>Go to the <SKLM_INSTALL_HOME>\migration\utilities\v2 directory and run the following command:</p> <pre>restoreV2.bat</pre> Linux <p>Go to the <SKLM_INSTALL_HOME>/migration/utilities/v2 directory and run the following command:</p> <pre>restoreV2.sh</pre> 4. Restart IBM Security Key Lifecycle Manager server. <p>Note: By using the migration restore script, you can restore roles, users, and groups from IBM Security Key Lifecycle Manager, Version 2.0 backup.</p>
---------------------------------	--

What to do next

Rollovers that are configured for LTO key groups and 3592 certificates are not automatically restored from the earlier versions of IBM Security Key Lifecycle Manager. You must manually set the rollover for certificates and key groups.

For more information, see “Restoring rollover certificates and key groups” on page 152.

Backing up IBM Security Key Lifecycle Manager, Version 2.0.1 (TKLM V2.0.1) data

Use the IBM Security Key Lifecycle Manager, Version 2.6 backup utility to create IBM Security Key Lifecycle Manager, Version 2.0.1 (formerly called Tivoli Key Lifecycle Manager, Version 2.0.1) backup files.

Before you begin

You must install IBM Security Key Lifecycle Manager, Version 2.6 on a system. Ensure that the system with IBM Security Key Lifecycle Manager, Version 2.0.1 with fix pack 5 is available. The keystore must be configured before you run the backup operation.

About this task

You can use the backup utility to create cross-platform backup files in a manner that is independent of operating systems and directory structure of the server. You can restore these cross-platform compatible backup files on a system with IBM Security Key Lifecycle Manager, Version 2.6 across operating systems.

Procedure

Run the followings steps on systems where the IBM Security Key Lifecycle Manager version 2.6 and version 2.0.1 are installed.

IBM Security Key Lifecycle Manager, Version 2.6	<div><div>1. Log on to the system with your user credentials.</div><div>2. Locate the backup utilities folder.</div><div><div>Windows</div><div><SKLM_INSTALL_HOME>\migration\utilities\v2</div><div>Default location is C:\Program Files (x86)\IBM\SKLMV26\migration\utilities\v2.</div><div><div>Linux</div><div><SKLM_INSTALL_HOME>/migration/utilities/v2</div><div>Default location is /opt/IBM/SKLMV26/migration/utilities/v2.</div></div></div><div>3. Copy the rt.jar and xml.jar files from <WAS_HOME>/java/jre/lib folder to the backup utilities folder that is described in Step b.</div></div>
--	---

IBM Security Key Lifecycle Manager, Version 2.0.1	<ol style="list-style-type: none"> 1. Log on to the system with your user credentials. 2. Copy v2 folder from the system where IBM Security Key Lifecycle Manager, Version 2.6 is installed to a local directory of your choice. 3. Edit backup.properties in the v2 folder to configure properties as shown in the following example. You must set values for all the properties, except for the BACKUP_DIR property (optional). If you do not specify the value for BACKUP_DIR, the backup file is created in the backup subfolder under the same directory from where you run the backup utility. Note: On Windows operating system, the backup.properties file that you use for backup operations must not contain the property keys and values with leading or trailing spaces. Windows TKLM_TIP_HOME=C:\\IBM\\tivoli\\tiptklmV2 DB_PASSWORD=tklmb2 KEYSTORE_PASSWORD=Passw0rd TIP_USER_PWD=tipadmin BACKUP_PASSWORD=passw0rd123 BACKUP_DIR=C:\\tklmv201_backup Linux TKLM_TIP_HOME=/opt/IBM/tivoli/tiptklmV2/ DB_PASSWORD=tklmb2 KEYSTORE_PASSWORD=Passw0rd TIP_USER_PWD=tipadmin BACKUP_PASSWORD=passw0rd123 BACKUP_DIR=/tklmv201_backup Note: On Windows operating system, when you specify path in the properties file, use either “/” or “\\” as path separator as shown in following example. C:\\tklmv201_backup Or C:/tklmv201_backup 4. Open a command prompt and run the backup utility. Windows Go to the v2 directory (see Step b) and run the following command: backupV2.bat Linux Go to the v2 directory (see Step b) and run the following command: backupV2.sh
--	---

What to do next

- Review the directory that contains backup files to ensure that the backup file exists. The backup files are created in the location that you specified for BACKUP_DIR in the backup.properties file.
- Check the backup.log file for errors or exceptions. The backup.log file is created in the same directory where you run the backup utility. For a successful backup operation, ensure that there are no errors or exceptions in the log file.
- Retain the backup password for future use in case you restore the backup.
- Do not edit a file in the backup archive. The file that you attempt to edit becomes unreadable.

Restoring IBM Security Key Lifecycle Manager, Version 2.0.1 (TKLM V2.0.1) backup files

You can restore the IBM Security Key Lifecycle Manager, Version 2.0.1 (formerly called Tivoli Key Lifecycle Manager, Version 2.0.1) cross-platform backup files on a system with IBM Security Key Lifecycle Manager, Version 2.6 by using the graphical user interface, command-line interface, REST interface, or the migration restore script.

Before you begin

Install IBM Security Key Lifecycle Manager, Version 2.6 on a system. You must have the backup file from the earlier version and ensure that you have the password that you used when the backup file was created.

Note: Your role must have the permissions to run backup and restore operations.

About this task

You can restore IBM Security Key Lifecycle Manager, Version 2.0.1 cross-platform compatible backup files on a system with IBM Security Key Lifecycle Manager, Version 2.6 across operating systems.

Before you start a restore task, isolate the system for maintenance. Take a backup of the existing system. You can later use this backup to bring the system back to original state if any issues occur during the restore process. You must restart the IBM Security Key Lifecycle Manager server immediately after the restore occurs. Verify the environment before you bring the IBM Security Key Lifecycle Manager server back into production.

Procedure

1. Log on to the system where IBM Security Key Lifecycle Manager, Version 2.6 is installed.
2. Copy the backup file, for example `tklm_v2.0.1.5_20150729013250-0400_migration_backup.jar`, from version 2.0.1 system to a directory of your choice.
3. Restore the backup file by using any of the following methods.

Graphical user interface	<ol style="list-style-type: none"> 1. Log on to the graphical user interface as an authorized user, for example, SKLMAdmin. 2. On the Welcome page, click Backup and Restore. 3. In the Backup repository location field, specify full path of the directory that contains the version 2.0.1 backup file. To locate the directory, click Browse. 4. Click Display Backups to display the backup files that you want to restore. 5. In the Backup and Restore table, select a backup file. 6. Click Restore From Backup. 7. On the Restore Backup page, specify the backup password that you used to create the backup file. 8. Click Restore Backup. 9. Restart IBM Security Key Lifecycle Manager server. <p>Note: By using the graphical user interface, you cannot restore roles, users, and groups from IBM Security Key Lifecycle Manager, Version 2.0.1 backup.</p>
Command-line interface	<ol style="list-style-type: none"> 1. Go to the WAS_HOME/bin directory. For example, <ul style="list-style-type: none"> Windows <pre>cd drive:\Program Files (x86)\IBM\WebSphere\AppServer\bin</pre> Linux <pre>cd /opt/IBM/WebSphere/AppServer/bin</pre> 2. Start the wsadmin interface by using an authorized user ID, such as SKLMAdmin. For example, <ul style="list-style-type: none"> Windows <pre>wsadmin -username SKLMAdmin -password mypwd -lang jython</pre> Linux <pre>./wsadmin.sh -username SKLMAdmin -password mypwd -lang jython</pre> 3. Run the tklmbBackupRunRestore CLI command by specifying the parameters such as the backup file name with its full path and backup password that you used to create the backup as shown in the following example. <pre>print AdminTask.tklmbBackupRunRestore ('[-backupFilePath /opt/mysklmbbackups/tklm_v2.0.1.5_20150729013250-0400_ -password myBackupPwd]')</pre> 4. Restart IBM Security Key Lifecycle Manager server. <p>Note: By using the command-line interface, you cannot restore roles, users, and groups from IBM Security Key Lifecycle Manager, Version 2.0.1 backup.</p>

REST interface	<ol style="list-style-type: none"> 1. Open a REST client. 2. Obtain a unique user authentication identifier to access IBM Security Key Lifecycle Manager REST services. For more information about the authentication process, see Authentication process for REST services. 3. To invoke Backup Run Restore REST Service, send the HTTP POST request with backup file name with its full path and backup password as parameters. Pass the user authentication identifier that you obtained in Step b along with the request message as shown in the following example. POST https://localhost:9080/SKLM/rest/v1/ckms/restore Content-Type: application/json Accept: application/json Authorization: SKLMAuth authId=139aeh34567m Accept-Language: en {"backupFilePath":"/opt/mysklmbackups/tklm_v2.0.1.5_20150729013250-0400_migr backup.jar","password":"myBackupPwd"} 4. Restart IBM Security Key Lifecycle Manager server. <p>Note: By using the REST interface, you cannot restore roles, users, and groups from IBM Security Key Lifecycle Manager, Version 2.0.1 backup.</p>
----------------	---

Migration restore script	<ol style="list-style-type: none"> 1. Locate the IBM Security Key Lifecycle Manager restore utilities. <ul style="list-style-type: none"> Windows <pre><SKLM_INSTALL_HOME>\migration\utilities\v2</pre> <p>Default location is C:\Program Files (x86)\IBM\SKLMV26\migration\utilities\v2.</p> Linux <pre><SKLM_INSTALL_HOME>/migration/utilities/v2</pre> <p>Default location is /opt/IBM/SKLMV26/migration/utilities/v2.</p> 2. Edit restore.properties in the v2 folder to configure properties as shown in the following example: <p>Note: On Windows operating system, the restore.properties file that you use for restore operations must not contain the property keys and values with leading or trailing spaces.</p> <ul style="list-style-type: none"> Windows <pre>WAS_HOME=C:\\Program Files (x86)\\IBM\\WebSphere\\AppServer BACKUP_PASSWORD=passw0rd123 DB_PASSWORD=sk1mdb26 RESTORE_FILE=C:\\tklmv201_restore\\tklm_v2.0.1.5_20150729013250-0400_r WAS_USER_PWD=wasadmin RESTORE_USER_ROLES=y</pre> Linux <pre>WAS_HOME=/opt/IBM/WebSphere/AppServer BACKUP_PASSWORD=passw0rd123 DB_PASSWORD=sk1mdb26 RESTORE_FILE=/tklmv201_restore/tklm_v2.0.1.5_20150729013250-0400_r WAS_USER_PWD=wasadmin RESTORE_USER_ROLES=y</pre> <p>Note: On Windows operating system, when you specify path in the properties file, use either “/” or “\\” as path separator as shown in the following example.</p> <pre>C:\\tklmv201_restore</pre> <p>Or</p> <pre>C:/tklmv201_restore</pre> 3. Open a command prompt and run the restore utility. <ul style="list-style-type: none"> Windows <p>Go to the <SKLM_INSTALL_HOME>\migration\utilities\v2 directory and run the following command:</p> <pre>restoreV2.bat</pre> Linux <p>Go to the <SKLM_INSTALL_HOME>/migration/utilities/v2 directory and run the following command:</p> <pre>restoreV2.sh</pre> 4. Restart IBM Security Key Lifecycle Manager server. <p>Note: By using the migration restore script, you can restore roles, users, and groups from IBM Security Key Lifecycle Manager, Version 2.0.1 backup.</p>
---------------------------------	---

What to do next

Rollovers that are configured for LTO key groups and 3592 certificates are not automatically restored from the earlier versions of IBM Security Key Lifecycle Manager. You must manually set the rollover for certificates and key groups.

For more information, see “Restoring rollover certificates and key groups” on page 152.

Backing up IBM Security Key Lifecycle Manager, Version 2.5 data

Use the IBM Security Key Lifecycle Manager, Version 2.6 backup utility to create IBM Security Key Lifecycle Manager, Version 2.5 cross-platform backup files.

Before you begin

You must install IBM Security Key Lifecycle Manager, Version 2.6 on a system. Ensure that the system with IBM Security Key Lifecycle Manager, Version 2.5 with fix pack 3 is available.

About this task

You can use the backup utility to create cross-platform backup files in a manner that is independent of operating systems and directory structure of the server. You can restore these cross-platform compatible backup files on a system with IBM Security Key Lifecycle Manager, Version 2.6 across operating systems.

Procedure

Run the followings steps on systems where the IBM Security Key Lifecycle Manager version 2.6 and version 2.5 are installed.

IBM Security Key Lifecycle Manager, Version 2.6	<div>1. Log on to the system with your user credentials.</div> <div>2. Locate the backup utilities folder.</div> <div>Windows <div><SKLM_INSTALL_HOME>\migration\utilities\sklmv25</div><div>Default location is C:\Program Files (x86)\IBM\SKLMV26\migration\utilities\sklmv25.</div></div> <div>Linux <SKLM_INSTALL_HOME>/migration/utilities/sklmv25 <div>Default location is /opt/IBM/SKLMV26/migration/utilities/sklmv25.</div></div> <div>3. Copy the rt.jar and xml.jar files from <WAS_HOME>/java/jre/lib folder to the backup utilities folder that is described in Step b.</div>
--	--

IBM Security Key Lifecycle Manager, Version 2.5	<ol style="list-style-type: none"> 1. Log on to the system with your user credentials. 2. Copy sklmv25 folder from the system where IBM Security Key Lifecycle Manager, Version 2.6 is installed to a local directory of your choice. 3. Edit backup.properties in the sklmv25 folder to configure properties as shown in the following example. You must set values for all the properties, except for the BACKUP_DIR property (optional). If you do not specify the value for BACKUP_DIR, the backup file is created in the backup subfolder under the same directory from where you run the backup utility. Note: On Windows operating system, the backup.properties file that you use for backup operations must not contain the property keys and values with leading or trailing spaces. Windows WAS_HOME=C:\\Program Files (x86)\\IBM\\WebSphere\\AppServer BACKUP_PASSWORD=passw0rd123 DB_PASSWORD=sklmb2 WAS_USER_PWD=wasadmin BACKUP_DIR=C:\\sklmv25_backup Linux WAS_HOME=/opt/IBM/WebSphere/AppServer BACKUP_PASSWORD=passw0rd123 DB_PASSWORD=sklmb2 WAS_USER_PWD=wasadmin BACKUP_DIR=/sklmv25_backup Note: On Windows operating system, when you specify path in the properties file, use either "/" or "\\" as path separator as shown in following example. C:\\sklmv25_backup Or C:/sklmv25_backup 4. Open a command prompt and run the backup utility. Windows Go to the sklmv25 directory (see Step b) and run the following command: backupV25.bat Linux Go to the sklmv25 directory (see Step b) and run the following command: backupV25.sh
--	--

What to do next

- Review the directory that contains backup files to ensure that the backup file exists. The backup files are created in the location that you specified for BACKUP_DIR in the backup.properties file.
- Check the backup.log file for errors or exceptions. The backup.log file is created in the same directory where you run the backup utility. For a successful backup operation, ensure that there are no errors or exceptions in the log file.
- Retain the backup password for future use in case you restore the backup.
- Do not edit a file in the backup archive. The file that you attempt to edit becomes unreadable.

Restoring IBM Security Key Lifecycle Manager, Version 2.5 backup files

You can restore the IBM Security Key Lifecycle Manager, Version 2.5 cross-platform backup files on a system with IBM Security Key Lifecycle Manager, Version 2.6 by using the graphical user interface, command-line interface, REST interface, or the migration restore script.

Before you begin

Install IBM Security Key Lifecycle Manager, Version 2.6 on a system. You must have the backup file from the earlier version and ensure that you have the password that you used when the backup file was created.

Note: Your role must have the permissions to run backup and restore operations.

About this task

You can restore IBM Security Key Lifecycle Manager, Version 2.5 cross-platform compatible backup files on a system with IBM Security Key Lifecycle Manager, Version 2.6 across operating systems.

Before you start a restore task, isolate the system for maintenance. Take a backup of the existing system. You can later use this backup to bring the system back to original state if any issues occur during the restore process. You must restart the IBM Security Key Lifecycle Manager server immediately after the restore occurs. Verify the environment before you bring the IBM Security Key Lifecycle Manager server back into production.

Procedure

- 1. Log on to the system where IBM Security Key Lifecycle Manager, Version 2.6 is installed.
- 2. Copy the backup file, for example sklm_v2.5.0.3_20150729013250-0400_migration_backup.jar, from version 2.5 system to a directory of your choice.
- 3. Restore the backup file by using any of the following methods.

Graphical user interface	<ul style="list-style-type: none">1. Log on to the graphical user interface as an authorized user, for example, SKLMAdmin.2. On the Welcome page, click Backup and Restore.3. In the Backup repository location field, specify full path of the directory that contains the version 2.5 backup file. To locate the directory, click Browse.4. Click Display Backups to display the backup files that you want to restore.5. In the Backup and Restore table, select a backup file.6. Click Restore From Backup.7. On the Restore Backup page, specify the backup password that you used to create the backup file.8. Click Restore Backup.9. Restart IBM Security Key Lifecycle Manager server. <p>Note: By using the graphical user interface, you cannot restore roles, users, and groups from IBM Security Key Lifecycle Manager, Version 2.5 backup.</p>
---------------------------------	--

Command-line interface	<ol style="list-style-type: none"> Go to the WAS_HOME/bin directory. For example, Windows <code>cd drive:\Program Files (x86)\IBM\WebSphere\AppServer\bin</code> Linux <code>cd /opt/IBM/WebSphere/AppServer/bin</code> Start the wsadmin interface by using an authorized user ID, such as SKLMAdmin. For example, Windows <code>wsadmin -username SKLMAdmin -password mypwd -lang jython</code> Linux <code>./wsadmin.sh -username SKLMAdmin -password mypwd -lang jython</code> Run the tklmBackupRunRestore CLI command by specifying the parameters such as the backup file name with its full path and backup password that you used to create the backup as shown in the following example. <pre>print AdminTask.tklmBackupRunRestore ('[-backupFilePath /opt/mysklmbackups/sklm_v2.5.0.3_20150729013250-0400_mig -password myBackupPwd]')</pre> Restart IBM Security Key Lifecycle Manager server. <p>Note: By using the command-line interface, you cannot restore roles, users, and groups from IBM Security Key Lifecycle Manager, Version 2.5 backup.</p>
REST interface	<ol style="list-style-type: none"> Open a REST client. Obtain a unique user authentication identifier to access IBM Security Key Lifecycle Manager REST services. For more information about the authentication process, see Authentication process for REST services. To invoke Backup Run Restore REST Service, send the HTTP POST request with backup file name with its full path and backup password as parameters. Pass the user authentication identifier that you obtained in Step b along with the request message as shown in the following example. <pre>POST https://localhost:9080/SKLM/rest/v1/ckms/restore Content-Type: application/json Accept: application/json Authorization: SKLMAuth authId=139aeh34567m Accept-Language: en {"backupFilePath":"/opt/mysklmbackups/sklm_v2.5.0.3_20150729013250-0400_mig backup.jar","password":"myBackupPwd"}</pre> Restart IBM Security Key Lifecycle Manager server. <p>Note: By using the REST interface, you cannot restore roles, users, and groups from IBM Security Key Lifecycle Manager, Version 2.5 backup.</p>

Migration restore script	<ol style="list-style-type: none"> Locate the IBM Security Key Lifecycle Manager restore utilities. <ul style="list-style-type: none"> Windows <pre><SKLM_INSTALL_HOME>\migration\utilities\sklmv25</pre> <p>Default location is C:\Program Files (x86)\IBM\SKLMV26\migration\utilities\sklmv25.</p> Linux <pre><SKLM_INSTALL_HOME>/migration/utilities/sklmv25</pre> <p>Default location is /opt/IBM/SKLMV26/migration/utilities/sklmv25.</p> Edit restore.properties in the sklmv25 folder to configure properties as shown in the following example. <p>Note: On Windows operating system, the restore.properties file that you use for restore operations must not contain the property keys and values with leading or trailing spaces.</p> <ul style="list-style-type: none"> Windows <pre>WAS_HOME=C:\\Program Files (x86)\\IBM\\WebSphere\\AppServer BACKUP_PASSWORD=passw0rd123 DB_PASSWORD=sklmdb26 RESTORE_FILE=C:\\sklmv25_restore\\sklm_v2.5.0.3_20150729013250-0400_mi WAS_USER_PWD=wasadmin</pre> Linux <pre>WAS_HOME=/opt/IBM/WebSphere/AppServer BACKUP_PASSWORD=passw0rd123 DB_PASSWORD=sklmdb26 RESTORE_FILE=/sklmv25_restore/sklm_v2.5.0.3_20150729013250-0400_migrat WAS_USER_PWD=wasadmin</pre> <p>Note: On Windows operating system, when you specify path in the properties file, use either "/" or "\\" as path separator as shown in the following example.</p> <pre>C:\\sklmv25_restore</pre> <p>Or</p> <pre>C:/sklmv25_restore</pre> Open a command prompt and run the restore utility. <ul style="list-style-type: none"> Windows <p>Go to the <SKLM_INSTALL_HOME>\migration\utilities\sklmv25 directory and run the following command:</p> <pre>restoreV25.bat</pre> Linux <p>Go to the <SKLM_INSTALL_HOME>/migration/utilities/sklmv25 directory and run the following command:</p> <pre>restoreV25.sh</pre> Restart IBM Security Key Lifecycle Manager server. <p>Note: By using the migration restore script, you can restore roles, users, and groups from IBM Security Key Lifecycle Manager, Version 2.5 backup.</p>
---------------------------------	--

Restoring rollover certificates and key groups

Rollovers that are configured for LT0 key groups and 3592 certificates are not automatically restored from the earlier versions of IBM Security Key Lifecycle Manager. To specify these rollovers manually in version 2.6, use the rollover file `scheduledTasks.txt`, which is created in the <WAS_HOME>/products/sklm/config directory during restore process.

Procedure

1. Open a command prompt.
2. Go to the following directory.

Windows

`<SKLM_INSTALL_HOME>\migration\bin`

Linux `<SKLM_INSTALL_HOME>/migration/bin`

3. Run the following command.

Windows

Run the **recreatetask.bat** utility.

```
recreatetask.bat <WAS_HOME> <SKLMADMIN_USER> <SKLMADMIN_PASSWD>  
<SKLM_HOME>\config\scheduledTasks.txt <Logfile> <SKLM_INSTALL_HOME>
```

Linux Run the **recreatetask.sh** utility.

```
recreatetask.sh <WAS_HOME> <SKLMADMIN_USER> <SKLMADMIN_PASSWD>  
<SKLM_HOME>/config/scheduledTasks.txt <Logfile> <SKLM_INSTALL_HOME>
```

Where, <Logfile> is the log file name to which the log information is written, for example:

`C:\Program Files (x86)\IBM\WebSphere\AppServer\products\sklm\logs\rolloverlogs.txt`

<SKLMADMIN_USER> and <SKLMADMIN_PASSWD> are IBM Security Key Lifecycle Manager administrator user ID and the password.

For the definitions of <WAS_HOME>, <SKLM_HOME>, and <SKLM_INSTALL_HOME>, see Definitions for *HOME* and other directory variables.

Key loss prevention

To prevent loss of encryption data for mission-critical devices and keys, always maintain a minimum of two instances of IBM Security Key Lifecycle Manager. Ensure that one of the instances is a replica of the same devices and keys. You might provide more than two redundant instances.

IBM Security Key Lifecycle Manager provides support for DS5000 storage servers that automatically generates keys when a new DS5000 device is registered in IBM Security Key Lifecycle Manager.

Do not use a setting of 1 (auto accept) for the DS5000 device family. This setting allows generation and serving of keys to DS5000 storage servers before you backup data. For all other device families, back up any new keys that are served.

Remove the backup files from the server and store in a secure location. For example, copy the backup files to a CD/DVD and lock in a safe place.

Note: Do not copy the files to an encrypted storage that is dependent on this product. Doing so might result in the backup not being available because the product is not available.

IBM Security Key Lifecycle Manager also provides these key loss options:

backup.keycert.before.serving

Set this property in the SKLMConfig.properties file to prevent serving new keys until the keys are backed up.

Automated backup script

Use the autobackup.bat script to automatically back up files. IBM Security Key Lifecycle Manager does not serve keys or certificates that are not

backed up if the value of the **backup.keycert.before.serving** property is set to true, or, is not present, in the SKLMConfig.properties file.

Configuring automated backup script

You might use the automated backup script to back up files.

About this task

IBM Security Key Lifecycle Manager does not serve keys or certificates that are not backed up if the value of the **backup.keycert.before.serving** property is set to true, or, is not present, in the SKLMConfig.properties file.

The automated backup script initiates a backup by calling these commands:

- **klmBackupIsRunning** to check whether a backup operation is running.
- **tklmBackupIsNeeded** or **Backup Is Needed REST Service** to determine whether new keys or certificates exist, but a backup is not yet run.
- **tklmBackupRun** or **Backup Run REST Service** to run the backup task.

Before you begin, determine the password that is used to encrypt the data in the backup file.

Procedure

1. Locate the script in this directory:

Windows

drive:\Program Files (x86)\IBM\SKLMV25\bin\samples\autobackup.bat

Linux and AIX®

path/IBM/SKLMV25/bin/samples/autobackup.sh

2. At the top of the autobackup.bat or autobackup.sh file, locate the lines that you change:

```
rem #####
rem #
rem #      EDIT THE PARAMETER VALUE IN THIS SECTION
rem #
rem tiphome : required, home directory of Tivoli Integrated Portal
rem username : required, username of the Tivoli Key Lifecycle Manager
rem user with klmBackup permission
rem password : required, password for the Tivoli Key Lifecycle Manager
rem user to log in
rem backuppw : required, password used for backup operation
rem backupdes : optional, description of the Tivoli Key Lifecycle
rem Manager backup
rem backupdir : optional, full path to the directory, where the
rem backup jar file is stored
rem backupDBdir : optional, full path to the directory, where the
rem database backup is stored
Set tiphome=
Set username=
Set password=
Set backuppw=
Set backupdes=
Set backupdir=
Set backupDBdir=
rem #####
```

3. Change the required lines in the script:

tiphome

Required. The WebSphere Application Server home directory.

For example:

Set `tiphome=C:/Progra~2/IBM/WebSphere/AppServer`

username

Required. A user ID that has **k1mBackup** permission. Use this user ID to log in to IBM Security Key Lifecycle Manager. The user ID can also be an existing user ID such as SKLMAdmin.

password

Required. The password of the user ID that has **k1mBackup** permission.

backuppw

Required. A password that is used to encrypt the data in the backup file. The value can range between a minimum of 6 and a maximum of 32 characters.

You can use a different password for each backup file. When you restore a file, you must be able to provide the password that was used to encrypt the data in that file during the backup task.

backupdes

Optional. More information about the purpose or use of the backup file.

backupdir

Optional. A directory that stores the JAR files with backup data for IBM Security Key Lifecycle Manager. Specify the full path to the directory.

If the backup is successful, the value that you specify is written as the value of the **tk1m.backup.dir** property in the `SKLMConfig.properties` file.

Note:

- If you do not specify a value for this parameter and no successful backup was run before, the default is the `SKLM_HOME/backup` directory.
- If you specify a relative path (not suggested) such as `mybackupdir`, the backup is created in the `WAS_HOME/profiles/KLMProfile/mybackupdir` directory.
- IBM Security Key Lifecycle Manager can create a backup file in any directory for which the operating system superuser has permission to write the file. The superuser is Administrator on Windows systems or root on systems such as Linux or AIX.
- Do not create the backup file in the same directory that contains the database backup.

backupDBdir

Optional. A directory in the IBM Security Key Lifecycle Manager database that contains temporary backup data for IBM Security Key Lifecycle Manager. If no parameter is specified, the directory that is used is the value of the **tk1m.backup.db2.dir** property in the `datastore.properties` file. The file is located in the `WAS_HOME\products\sklm\config` directory, or a temporary system directory if the directory specified by the **tk1m.backup.db2.dir** property does not exist.

4. Run the script:

- Immediately. Type:

Windows

`drive:\Program Files (x86)\IBM\SKLMV25\bin\samples\autobackup.bat`

Linux and AIX

`path/IBM/SKLMV25/bin/samples/autobackup.sh`

- On a scheduled basis.

Depending on the operating system, enable the script in a cron job or by using the Windows Scheduler.

KMIP objects management

The IBM Security Key Lifecycle Manager server supports Key Management Interoperability Protocol (KMIP) communication with client devices. You can create and manage cryptographic objects by using a set of operations that IBM Security Key Lifecycle Manager provides.

You can use IBM Security Key Lifecycle Manager graphical user interface for the following cryptographic object management activities:

- Registering client devices with the server
- Creating and configuring cryptographic objects for the registered client devices
- Viewing objects for the registered client devices
- Modifying client device information by adding more objects or associating a new certificate
- Deleting client devices and the associated objects from the server
- Searching for objects that are managed by the server

Registering KMIP-compliant client devices

You must register KMIP-compliant client devices with the IBM Security Key Lifecycle Manager server before the client communicates with server for key management operations.

About this task

Use **Client Dashboard** to manage client devices and its objects. You can use the dashboard to view, register, modify, and delete client devices and the associated cryptographic objects.

Associate a certificate with client device for secure communication with the server. Before you register the client, determine which of the following client device certificates to be used for communication:

- Existing client device certificate that is not in use.
- Accepting a pending client device certificate.
- Importing a client device certificate.

You can also register client device without an associated certificate. You can later associate by selecting certificate from the pending certificate list. Click the **Pending client registration requests** link on the dashboard to select the certificate. If you accept, the certificate is imported into the database and marked as trusted. The certificate can then be used for secure communication between the client device and IBM Security Key Lifecycle Manager. You can also associate a certificate when you modify client device information.

Procedure

1. Log on to the graphical user interface by using your credentials.
2. On the Welcome page, click **Clients and Groups**.
3. On **Client Dashboard**, click **Create**.
4. In the **Client Name** field, specify a name for the client.
5. Select a client certificate for secure communication with the server.

None	The client device is registered with the server without an associated client communication certificate.
Use existing client certificate not in use	Use an existing client certificate in the database, which is not in use by any other client devices. Select the certificate from the drop-down list.
Accept pending client certificate	Select a certificate from the pending certificate list that is pushed to the server from a client device and is yet to be accepted for communication with the server. Use the following steps to accept the client communication certificate and mark it as trusted when you register the client device: <ol style="list-style-type: none">1. Specify a name for the certificate in the Certificate name field.2. Select a certificate from the drop-down list.
Import client certificate	Import a client device certificate into IBM Security Key Lifecycle Manager for secure communication with the client you are registering. <ol style="list-style-type: none">1. Specify a name for the certificate in the Certificate name field.2. Click Browse to select the file and import.

6. Click **Register Client**.

What to do next

Associate cryptographic objects with the registered client. See “Creating cryptographic objects for a client device.”

Creating cryptographic objects for a client device

Create and configure cryptographic objects for the KMIP-compliant client device that you registered with IBM Security Key Lifecycle Manager.

About this task

Use **Client Dashboard** to manage client devices and its objects. You can use the dashboard to view, register, modify, and delete client devices and the associated cryptographic objects.

Use the Objects with Client page to create and configure the following objects for the client device that you registered:

- Symmetric keys
- Key pairs

Procedure

1. Log on to the graphical user interface by using your credentials.
2. On the Welcome page, click **Clients and Groups**.

3. On **Client Dashboard**, click **Create**.
4. Create and register a client device by using the **Register Client** tab. See “Registering KMIP-compliant client devices” on page 156.
5. In the **Add Objects** tab, add and configure objects for the client device:

None	Indicates that the client device is not associated with any of the objects.
Symmetric key	<p>Create the symmetric key object with the following configuration settings:</p> <ul style="list-style-type: none"> • Number of symmetric keys for the client device. • Cryptographic algorithm that the object uses for data encryption or decryption, such as AES or 3DES. • Bit length of the symmetric key object. • Prefix for the key. You must specify a three character value for the key by using the alphabetic characters. • Cryptographic usage mask that defines the cryptographic functions to be performed by using the object, such as Encrypt, Decrypt, Encrypt Decrypt, Sign, Sign Verify, Verify, Wrap, Unwrap, or Wrap Unwrap.
Key Pair	<p>Create the asymmetric key pair object with the following configuration settings:</p> <ul style="list-style-type: none"> • Number of key pair objects for the client device. • Cryptographic algorithm that the object uses for data encryption or decryption, such as RSA or DSA. • Prefix for the key. You must specify a three character value for the key by using the alphabetic characters. • Cryptographic usage mask that defines the cryptographic functions to be performed by using the object, such as Encrypt, Decrypt, Encrypt Decrypt, Sign, Sign Verify, Verify, Wrap, Unwrap, or Wrap Unwrap.

6. Click **Save and Exit**. To save and add more objects, click **Save and Add More Objects**.

Modifying client device information

Use the Modify Client page to modify client device information to suit your changing needs. You can add more objects and associate a certificate with the client device that is registered with IBM Security Key Lifecycle Manager.

About this task

Use the Client Dashboard page to manage client devices and its objects. You can use the dashboard to view, register, modify, and delete client devices and the associated cryptographic objects.

Your role must have a permission to the modify action.

Procedure

1. Log on to the graphical user interface by using your credentials.
2. On the Welcome page, click **Clients and Groups**.
3. Select a client device from the **Client Name** column.

4. Click **Modify**.
5. Alternatively, right-click a client name and then select **Modify**, or double-click a client entry.
6. On the Modify Client dialog, add more objects or associate a certificate to the client device to fit your needs.
7. Click **Save and Exit**. To save and add more objects, click **Save and Add More Objects**.

Deleting client device information

You can delete client device and its objects from the IBM Security Key Lifecycle Manager database if they are no longer required.

About this task

Use the Client Dashboard page to manage client devices and its objects. You can use the dashboard to view, register, modify, and delete client devices and the associated cryptographic objects.

Your role must have a permission to the delete action.

Before you begin, ensure that a current backup exists for the IBM Security Key Lifecycle Manager database.

Procedure

1. Log on to the graphical user interface by using your credentials.
2. On the Welcome page, click **Clients and Groups**.
3. Select a client device from the **Client Name** column.
4. Click **Delete**.
5. Alternatively, right-click a client name and then select **Delete**, or double-click a client entry.
6. On the Confirm dialog, read the confirmation message before you delete the client device. Click **OK**. The client device and its objects are removed from the IBM Security Key Lifecycle Manager database.

Hardware Security Module usage in IBM Security Key Lifecycle Manager

You must add the parameters to the IBM Security Key Lifecycle Manager configuration file to define a Hardware Security Module (HSM).

You can use HSM for storing master key to protect all passwords that are stored in the IBM Security Key Lifecycle Manager database. You can enable this capability for the installations with existing data, or for the new installations of IBM Security Key Lifecycle Manager.

IBM Security Key Lifecycle Manager supports the following cryptography cards:

- SafeNet Luna SA 4.5
- SafeNet Luna SA 5.0
- nCipher nShield Connect 1500
- IBM 4765 PCIe Cryptographic Coprocessor

Note:

- You can use SafeNet Luna SA 4.5, SafeNet Luna SA 5.0, and IBM 4765 PCIe Cryptographic Coprocessor only when the keystore is not defined in IBM Security Key Lifecycle Manager. These cards do not allow import of keys from outside.
- IBM 4765 PCIe Cryptographic Coprocessor is supported only for the following PKCS#11 crypto operations:
 - Translate an AES 128-bit or 256-bit software key to an AES hardware (PKCS#11) key
 - Generate an AES 128-bit or 256-bit key
 - Encrypt and decrypt data by using an AES key and an AES/ECB/NoPadding cipher
 - Store and retrieve an AES key to/from a PKCS11IMPLKS (PKCS#11) keystore

You can use the following configuration parameters to define HSM:

- **pkcs11.pin**
- **pkcs11.pin.obfuscated**
- **pkcs11.config**

For HSM configuration parameter details, see the Reference topics in the IBM Security Key Lifecycle Manager documentation.

Sample HSM configuration files

Sample HSM configuration file for SafeNet Luna SA 4.5 and SafeNet Luna SA 5.0

```
#SafeNet Luna
name = TKLM
library=C:/Program Files/LunaSA/cryptoki.dll
description=Luna sample config

slotListIndex = 0

attributes (*, CKO_PRIVATE_KEY, *) = {
    CKA_SENSITIVE = true
}
attributes (GENERATE, CKO_SECRET_KEY, *) = {
    CKA_SENSITIVE = true
    CKA_ENCRYPT = true
    CKA_DECRYPT = true
}
attributes (IMPORT, CKO_PUBLIC_KEY, *) = {
    CKA_VERIFY = true
}
```

Note: For the **name** parameter, you must always specify the value TKLM.

Sample HSM configuration file for nCipher nShield Connect 1500

```
# nCipher nShield, nForce 4000 - Generation 2 cards
name = TKLM
library=C:/nCipher/nfast/cknfast.dll
description= nCipher sample config for TKLM

slotListIndex=1

attributes(*, CKO_SECRET_KEY, *) = {
    CKA_ENCRYPT=true
    CKA_DECRYPT=true
    CKA_SENSITIVE=true
    CKA_TOKEN=true
}
```

```

attributes(*, CKO_PRIVATE_KEY, *) = {
    CKA_SIGN=true
    CKA_SENSITIVE=false
    # CKA_DERIVE=true
    # when using KeyAgreement CKA_DERIVE should
    # set to true and CKA_SIGN should set to false
}

attributes(GENERATE, CKO_PUBLIC_KEY, *) = {
    CKA_VERIFY=true
}

attributes(GENERATE, CKO_PRIVATE_KEY, CKK_RSA) = {
    CKA_DECRYPT=true
    CKA_UNWRAP=true
    CKA_EXTRACTABLE=true
}

attributes(*, CKO_PUBLIC_KEY, CKK_RSA) = {
    CKA_ENCRYPT=true
    CKA_WRAP=true
    CKA_VERIFY=true
}

attributes(IMPORT, CKO_PRIVATE_KEY, CKK_RSA) = {
    CKA_EXTRACTABLE=true
    CKA_DECRYPT=true
    CKA_UNWRAP=true
    CKA_DERIVE=true
}

```

Note: For the **name** parameter, you must always specify the value TKLM.

Configuring HSM parameters

You must use the **pkcs11.pin**, **pkcs11.pin.obfuscated**, and **pkcs11.config** configuration parameters to define HSM.

Procedure

1. Set up and configure the HSM as per the instructions from HSM manufacturers.
2. Add the **pkcs11.pin** and the **pkcs11.config** parameters to the IBM Security Key Lifecycle Manager configuration file. You can use the following CLI command or REST interface to add the parameter:

Command-line interface

```

print AdminTask.tklmConfigUpdateEntry('[-name pkcs11.pin -value
<hsm pin>]')

print AdminTask.tklmConfigUpdateEntry('[-name pkcs11.config -value
<hsm config file>]')

```

REST interface

```

PUT https://localhost:9080/SKLM/rest/v1/configProperties
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth authId=139aeh34567m
Accept-Language: en
{ "pkcs11.pin" : "<hsm pin>" }

PUT https://localhost:9080/SKLM/rest/v1/configProperties
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth authId=139aeh34567m
Accept-Language: en
{ "pkcs11.config" : "<hsm config file>" }

```

Note: *<hsm pin>* is the PIN for HSM. *<hsm config file>* is the full path and file name to the HSM configuration file. For example: C:\Program Files (x86)\IBM\WebSphere\AppServer\sklm\config\LunaSA.cfg

3. Restart IBM Security Key Lifecycle Manager.

Configuration requirements to use HSM

You must validate HSM installation with the tools that the HSM client provides after you install HSM as per the instructions from manufacturers. Only the 32-bit HSM client is supported.

- Perform the following steps to validate the HSM installation:
 - Create a symmetric key with **ckdemo** or **kSafe**.
kSafe is a tool that comes with the nCipher nShield Connect 1500 card.
ckdemo comes with the SafeNet Luna SA 4.5 card and SafeNet Luna SA 5.0 card.
 - List the key.
 - Delete the key.

- The nCipher nShield Connect 1500 card requires that the cknfastrc file contain the following configuration:

```
CKNFAST_OVERRIDE_SECURITY_ASSURANCES=import;
```

Note: If the cknfastrc file does not exist on your system, create the file and configure it. Save this file in the location that is mentioned in the HSM documentation.

- IBM Security Key Lifecycle Manager backup or replication does not back up the master key when it is placed in the HSM. To back up the HSM, follow the instructions in HSM documentation. You must back up the HSM because any master key loss might results in loss of all keys in IBM Security Key Lifecycle Manager.
- Use the SafeNet Luna SA 4.5 card and the SafeNet Luna SA 5.0 card only when the keystore is not defined in IBM Security Key Lifecycle Manager. These cards do not allow import of keys from outside.
- To clone IBM Security Key Lifecycle Manager, the HSM on the different systems must use the same master key. If you are using a network attached HSM, ensure that all your clients for HSM are pointing to the same area on the HSM network.

LDAP configuration

You can configure IBM Security Key Lifecycle Manager users in any of the LDAP repositories, such as IBM Security Directory Server or Microsoft Active Directory to access IBM Security Key Lifecycle Manager server.

You must add and configure LDAP user repository to the federated repository of WebSphere Application Server. IBM Security Key Lifecycle Manager uses application groups to enforce the role-based authorization for IBM Security Key Lifecycle Manager functions. For an IBM Security Key Lifecycle Manager user to run IBM Security Key Lifecycle Manager functions in an LDAP user repository, the user must be member of a specific IBM Security Key Lifecycle Manager application groups.

When you install IBM Security Key Lifecycle Manager, the application groups and users are created in a default file based repository in the WebSphere Application Server federated repository. When an LDAP user repository is added to the WebSphere Application Server federated repository, you must make LDAP user as

a member of IBM Security Key Lifecycle Manager application groups. You cannot make LDAP users as member of the groups in the default file based repository.

Cross repository group membership is not possible between a file-based repository and an LDAP repository. However, cross repository group membership is possible across an LDAP repository and a database-based repository. So, create a database-based repository and create all the IBM Security Key Lifecycle Manager application groups in this repository. The application groups that existed in file based repository are removed.

When the database-based repository is created and the IBM Security Key Lifecycle Manager application groups are added to this repository, the user in an LDAP repository can be made members of IBM Security Key Lifecycle Manager application groups in the database-based repository. Then, the user can log on to IBM Security Key Lifecycle Manager application and run IBM Security Key Lifecycle Manager application functions.

Prerequisites for LDAP integration

You might need to restore the following data to the state as before the LDAP configuration steps were run:

- WebSphere Application Server configuration data for IBM Security Key Lifecycle Manager
- IBM Security Key Lifecycle Manager application data

Run the following steps to back up the data:

1. Backup IBM Security Key Lifecycle Manager profile (KLMPProfile) in WebSphere Application Server:
 - a. In the WAS_HOME/bin directory, stop the WebSphere Application Server application.
 - b. Run the following command:

Windows

```
<WAS_HOME>\bin\manageProfiles.bat -backupProfile -profileName  
KLMPProfile -backupFile <path to a file>  
C:\Program Files (x86)\IBM\WebSphere\AppServer\bin\manageProfiles.bat  
backupProfile -profileName KLMPProfile -backupFile  
:\SKLM_WAS_ProfileBackup
```

Linux

```
<WAS_HOME>/bin/manageprofiles.sh -backupProfile -profileName  
KLMPProfile -backupFile <path to a file>  
/opt/IBM/WebSphere/AppServer/bin/manageprofiles.sh -backupProfile  
profileName KLMPProfile -backupFile /root/SKLM_WAS_ProfileBackup
```

- c. Start WebSphere Application Server.
2. Backup IBM Security Key Lifecycle Manager application data.
Use the graphical user interface, command-line interface, or REST interface to back up critical files for IBM Security Key Lifecycle Manager.

For more information about the **manageprofiles** command, see http://www-01.ibm.com/support/knowledgecenter/SSEQTP_8.5.5/com.ibm.websphere.base.doc/ae/rxml_manageprofiles.html.

Integrating LDAP with IBM Security Key Lifecycle Manager

You can configure IBM Security Key Lifecycle Manager users in any of the LDAP repositories, such as IBM Security Directory Server or Microsoft Active Directory to access IBM Security Key Lifecycle Manager server and call server APIs and CLIs.

Before you begin

For prerequisite information, see “LDAP configuration” on page 162

Procedure

1. Add LDAP repository to the federated repository. For the instructions, see “Adding LDAP repository to the federated repository.”
2. Create a data source from the WebSphere Integrated Solutions Console with jndi name jdbc/wimXADS. For the instructions, see “Creating a data source from WebSphere Integrated Solutions Console” on page 166.
3. Restart WebSphere Application Server.
4. Copy db2jcc.jar and db2jcc_license_cu.jar from the DB2SKLMV25 folder to the <WAS_HOME>/lib folder.

DB2SKLMV25 path:

Windows

C:\Program Files (x86)\IBM\DB2SKLMV25\java

Linux /opt/IBM/DB2SKLMV25/java

Default definition of <WAS_HOME> variable is typically:

Windows

C:\Program Files (x86)\IBM\WebSphere\AppServer

Linux /opt/IBM/WebSphere/AppServer

5. Create database-based repository to hold all the IBM Security Key Lifecycle Manager application groups. For the instructions, see “Creating a database-based repository” on page 167.
6. From WebSphere Integrated Solutions Console, add security role to user/group mapping and map administrator role to klmGUICLIAccessGroup . For the instructions, see “Adding security user roles from WebSphere Integrated Solutions Console” on page 168.
7. Restart WebSphere Application Server.
8. Add LDAP users to IBM Security Key Lifecycle Manager application groups. For the instructions, see “Adding LDAP users to IBM Security Key Lifecycle Manager application groups” on page 169
9. Take the IBM Security Key Lifecycle Manager application backup. The data in the database-based repository is also backed up.

What to do next

After LDAP is configured, you must run the subsequent tasks. For the details, see “Post-LDAP configuration tasks to support LDAP integration” on page 170

Adding LDAP repository to the federated repository

You must add LDAP repository to the federated repository to configure an LDAP repository, such as IBM Security Directory Server or Microsoft Active Directory in the federated repository.

About this task

For more information about configuring LDAP settings in a federated repository configuration, see http://www-01.ibm.com/support/knowledgecenter/api/redirect/wasinfo/v8r5/topic/com.ibm.websphere.base.doc/ae/twim_ldap_settings.html.

Procedure

1. Log on to WebSphere Integrated Solutions Console (<https://localhost:9083/ibm/console/logon.jsp>) as wasadmin user.
2. In the navigation bar, click **Security > Global security**.
3. Under User account repository, select **Federated repositories** from the **Available realm definitions** drop-down list.
4. Click **Configure**.
5. In the **Global security > Federated repositories** page, click **Add Repositories (LDAP, custom,etc...)**.
6. In the **Global security > Federated repositories > Repository reference** page, select **LDAP Repository** from the **New Repository** drop-down list.
7. In the **Global security > Federated repositories > Repository reference > New** page, specify name of the LDAP repository and other details according to your requirements.
8. Click **OK**.
9. Click **Save** to save the configuration.
10. In the **Global security > Federated repositories > Repository reference** page, specify the value for **Unique distinguished name of the base (or parent) entry in federated repositories**.
11. Click **OK**.
12. In the **Global security > Federated repositories** page, select the link to the LDAP repository that you created.
13. In the **Global security > Federated repositories > <LDAP Repository Name>** page, under Additional Properties, select **Federated repositories entity types to LDAP object classes mapping** link.
In the **Global security > Federated repositories > <LDAP Repository Name> > Federated repositories entity types to LDAP object classes mapping** page, ensure that each entity type listed is mapped to the correct object classes. Modify the values according to your requirements.
14. In the **Global security > Federated repositories** page, select the link to the LDAP repository that you created. Under Additional Properties, select **Group attribute definition**.
15. In the **Global security > Federated repositories > <LDAP Repository Name> > Group attribute definition** page, under Additional Properties, select **Member Attributes**.
16. In the **Global security > Federated repositories > <LDAP Repository Name> > Group attribute definition > Member attributes** page, ensure that uniqueMember member attribute is mapped to the correct object class. If this attribute is not present, create an attribute and map it to the correct object class.

What to do next

Create a data source from WebSphere Integrated Solutions Console.

Creating a data source from WebSphere Integrated Solutions Console

You must create a data source for the database-based repository to hold IBM Security Key Lifecycle Manager application groups. The database-based repository uses the tables that are created in the IBM Security Key Lifecycle Manager application database.

Procedure

1. Log on to Creating a data source from WebSphere Integrated Solutions Console (<https://localhost:9083/ibm/console/logon.jsp>) as a wasadmin user.
2. In the navigation bar, click **Resources > JDBC > Data sources**.
3. Select **Node=SKLMNode, Server-server1** from the **Select** drop-down list.
4. Click **New**.
5. In the Enter basic data source information dialog, specify the values.

Option	Description
Data source name	WIM Data Source
JNDI name	jdbc/wimXADS Note: Do not change this value.

6. Click **Next**.
7. In the Select JDBC provider dialog, select **Select an existing JDBC provider > SKLM XA DB2 JDBC Provider**.
8. Click **Next**.
9. In the Enter database specific properties for the data source dialog, specify the values:

Option	Description
Database name	SKLMDB26 or whatever the name that was used during IBM Security Key Lifecycle Manager installation.
Server name	localhost
Port number	50020 or whatever the port number that was used for DB2 during IBM Security Key Lifecycle Manager installation.

10. Click **Next**.
11. In the Setup security aliases dialog, select the values from the drop-down list:

Option	Description
Authentication alias for XA recovery	sklm_db
Component-managed authentication alias	sklm_db
Mapping-configuration alias	None
Container-managed authentication alias	sklm_db

12. Click **Next**.
13. Click **Finish** in the Summary page.
14. In the Data sources page, click **Save** to save the configuration.
15. Select **WIM Data Source** in the list of data sources.

16. Click **Test connection** to ensure that the connection test is successful.
17. Select the **WIM Data Source** link in the list of data sources.
18. In the **Data sources > WIM Data Source** page, click the **Custom properties** link.
19. Navigate to the pages to locate the link to the **webSphereDefaultIsolationLevel** property and click the link.
20. In the **Data sources > WIM Data Source > Custom properties > webSphereDefaultIsolationLevel** page, under the General Properties section, enter the value 2 in the **Value** field.
21. Click **OK**.
22. Click **Save** to save the configuration.

What to do next

Restart WebSphere Application Server.

Creating a database-based repository

Create a database-based repository to hold all the IBM Security Key Lifecycle Manager application groups and to remove all the IBM Security Key Lifecycle Manager application groups from file-based repository. You must add the IBM Security Key Lifecycle Manager application groups to database-based repository and update the WebSphere Application Server federated repository with LDAP repository.

Procedure

1. Go to the <WAS_HOME>/bin folder.

Note: All the .py python scripts are present in the <SKLM_HOME>/bin/LDAPIntegration directory.
<SKLM_HOME> path typically,

Windows

C:\Program Files (x86)\IBM\SKLMV26

Linux /opt/IBM/SKLMV26

2. Run the following commands:

```
wsadmin.bat -user <wasadmin user> -password <wasadmin passwd> -lang jython -f
<SKLM_HOME>\bin\LDAPIntegration\createDBRepos.py <WAS_HOME> <SKLM_DBNAME>
<SKLM_DBUSER> <SKLM_DBUSERPASSWD> <SKLM_DBPORT#>
```

Notes: On Linux platforms, use **wsadmin.sh** instead of **wsadmin.bat**

During IBM Security Key Lifecycle Manager installation, if you use the defaults,

```
SKLM_DBNAME = SKLMDB26
SKLM_DBUSER = sklmdb26
SKLM_DBPORT# = 50020
```

SKLM_DBUSERPASSWD is the IBM Security Key Lifecycle Manager database password that you specified during the installation.

3. Run the following command.

```
wsadmin.bat -user <wasadmin user> -password <wasadmin passwd> -lang jython -f
<SKLM_HOME>\bin\LDAPIntegration\removeGroupsFromDefRepos.py
```

4. From the WebSphere Integrated Solutions Console, modify Security role to user/group mapping for removing the administrator role mapping to klmGUICLIAccessGroup.
 - a. Log on to WebSphere Integrated Solutions Console (<https://localhost:9083/ibm/console/logon.jsp>).
 - b. In the navigation bar, click **Applications > Application Types > Application Types > WebSphere enterprise applications**.
 - c. Click the **sklm_kms** link.
 - d. In the **Enterprise Applications > sklm_kms** page, under the Detail Properties section, click the **Security role to user/group mapping** link.
 - e. In the **Enterprise Applications > sklm_kms > Security role to user/group mapping** page, select the **administrator** role.
 - f. Click **Map Groups**.
 - g. Select **klmGUICLIAccessGroup** from the list and click the left arrow button to remove **klmGUICLIAccessGroup** from the list.
 - h. Click **OK**.
 - i. Click the **Save** link to save the configuration.
5. Restart WebSphere Application Server
6. Run the following command.


```
wsadmin.bat -user <wasadmin user> -password <wasadmin passwd> -lang jython
-f <SKLM_HOME>\bin\LDAPIntegration\addGroupsToDBRepos.py
```
7. Run the following command.


```
wsadmin.bat -user <wasadmin user> -password <wasadmin passwd> -lang jython
-f <SKLM_HOME>\bin\LDAPIntegration\updateLDAPReposConfig.py <LDAPRepos Name
- name used earlier when LDAP repos was created>
```

What to do next

Add security role to user/group mapping and map administrator role to klmGUICLIAccessGroup.

Adding security user roles from WebSphere Integrated Solutions Console

You must add security role to user or group mapping, and map administrator role to klmGUICLIAccessGroup for integrating IBM Security Key Lifecycle Manager with LDAP user repositories.

About this task

Procedure

1. Log on to WebSphere Integrated Solutions Console (<https://localhost:9083/ibm/console/logon.jsp>) as a wasadmin user.
2. In the navigation bar, click **Applications > Application Types > Application Types > WebSphere enterprise applications**.
3. Click the **sklm_kms** link.
4. In the **Enterprise Applications > sklm_kms** page, under the Detail Properties section, click the **Security role to user/group mapping** link.
5. In the **Enterprise Applications > sklm_kms > Security role to user/group mapping** page, select the **administrator** role.
6. Click **Map Groups**.

7. In the **Enterprise Applications > sklm_kms > Security role to user/group mapping > Map users/groups** page:
 - a. Under the Search and Select Groups section, in the **Search string** text box, enter `klmGUICLIAccessGroup`.
 - b. Click **Search**.
 - c. Select `klmGUICLIAccessGroup` from the list and click the right arrow button. `klmGUICLIAccessGroup` is added to the **Selected** list.
 - d. Click **OK**.
 - e. Click **OK** in the **Enterprise Applications > sklm_kms > Security role to user/group mapping** page.
8. Click the **Save** link to save the configuration information.

What to do next

Restart WebSphere Application Server.

Adding LDAP users to IBM Security Key Lifecycle Manager application groups

You must add LDAP Users to IBM Security Key Lifecycle Manager Application Groups to integrate IBM Security Key Lifecycle Manager with LDAP user repositories.

Procedure

1. Go to the `<WAS_HOME>/bin` folder.

Note: All the .py python scripts are present in the `<SKLM_HOME>/bin/LDAPIntegration` directory.
`<SKLM_HOME>` path typically,

Windows

`C:\Program Files (x86)\IBM\SKLMV25`

Linux

`/opt/IBM/SKLMV25`

2. Run the following commands:

```
wsadmin.bat -user <wasadmin user> -password <wasadmin passwd> -lang jython -f
addLDAPUserToGroup.py <user uniqueName> <group name>
```

Notes: On Linux platforms, use **wsadmin.sh** instead of **wsadmin.bat**

The user unique name is the Unique Name component in LDAP registry. For example:

```
uid=001,c=in,ou=bluepages,o=ibm.com
```

For an LDAP user who needs IBM Security Key Lifecycle Manager admin access, the user must be made member of `klmGUICLIAccessGroup` and `klmSecurityOfficerGroup`. Run the following command:

```
wsadmin.bat -user <wasadmin user> -password <wasadmin passwd> -lang jython -f
<SKLM_HOME>\bin\LDAPIntegration\addLDAPUserToGroup.py <user uniqueName>
klmGUICLIAccessGroup
```

What to do next

Take IBM Security Key Lifecycle Manager application backup.

Post-LDAP configuration tasks to support LDAP integration

After LDAP configuration, you might need to complete extra tasks to ensure successful integration of IBM Security Key Lifecycle Manager with LDAP user repositories.

Important notes after the LDAP configuration

1. After the LDAP configuration, sklmadmin user that existed in the default file-based user repository cannot access IBM Security Key Lifecycle Manager application.
2. After the LDAP configuration, you must use **wsadmin** commands to create groups and to assign IBM Security Key Lifecycle Manager roles. You cannot use WebSphere Integrated Solutions Console. Run the following steps to add a group and assign a role to the group:
 - a. Go to `<WAS_HOME>/bin`.
 - b. Log on to wsadmin by using the following command:

```
wsadmin.bat -user <wasadmin user> -password <wasadmin passwd> -lang jython
```
 - c. To create a group and assign the role, run the following command:

```
AdminTask.createGroup<'[-cn <groupname> -parent "o=sklmrepdb.ibm"]'>
AdminTask.mapGroupsToAdminRole<'[-roleName <role> -groupids <groupname>]'>
```
3. After the LDAP configuration, you might want to restore the IBM Security Key Lifecycle Manager configuration in WebSphere Application Server to the state as before the LDAP configuration. To restore the configuration, run the following steps:
 - a. Stop WebSphere Application Server.
 - b. Stop WebSphere Application Server related processes, if any.
 - c. Restore WebSphere Application Server profile configuration that was taken before the LDAP configuration:
 - 1) Manually delete the KLMPProfile folder at `<WAS_HOME>/profiles/ KLMPProfile`.
 - 2) Run the **-validateAndUpdateRegistry** option of the **manageProfiles** command.

Windows

```
<WAS_HOME>\bin\manageProfiles.bat
-validateAndUpdateRegistry
```

For example: `C:\Program Files (x86)\IBM\WebSphere\AppServer\bin\manageProfiles.bat -validateAndUpdateRegistry`

Linux `<WAS_HOME>/bin/manageprofiles.sh -validateAndUpdateRegistry`

For example: `/opt/IBM/WebSphere/AppServer/bin/manageprofiles.sh -validateAndUpdateRegistry`

- 3) Restore the profile:

Windows

```
<WAS_HOME>\bin\manageProfiles.bat -restoreProfile
-backupFile <path to profile backup file>
```

For example: `C:\Program Files (x86)\IBM\WebSphere\AppServer\bin\manageProfiles.bat -restoreProfile -backupFile C:\SKLM_WAS_ProfileBackup`

Linux `<WAS_HOME>/bin/manageprofiles.sh -restoreProfile
-backupFile <path to profile backup file>`

For example: `/opt/IBM/WebSphere/AppServer/bin/
manageprofiles.sh -restoreProfile -backupFile
/root/SKLM_WAS_ProfileBackup`

For information about the **manageProfiles** command, see
[http://www-01.ibm.com/support/knowledgecenter/
SSEQTP_8.5.5/com.ibm.websphere.base.doc/ae/
rxml_manageprofiles.html](http://www-01.ibm.com/support/knowledgecenter/SSEQTP_8.5.5/com.ibm.websphere.base.doc/ae/rxml_manageprofiles.html).

- 4) Start WebSphere Application Server.
- 5) Restore IBM Security Key Lifecycle Manager backup that was taken before the LDAP configuration, if needed.
4. You must not restore IBM Security Key Lifecycle Manager application backup that is taken before the LDAP configuration after the LDAP configuration is done unless Step 3 in the **Important notes after the LDAP configuration** section is followed.
5. After the LDAP configuration, the tables are created in the IBM Security Key Lifecycle Manager database for the database-based repository. The IBM Security Key Lifecycle Manager groups are stored in these tables. If the IBM Security Key Lifecycle Manager server is configured for the replication and the replication happens to the configured clones, the groups in the database-based repository are also replicated on the clone. This is because the database tables of the database-based repository are also replicated to the clones.
6. If the IBM Security Key Lifecycle Manager server (master) that is configured to integrate with LDAP repositories and replication is enabled, when replication happens to the configured clones where LDAP is not configured, you can configure LDAP on the clone or not. If LDAP configuration must be done on the clone, run the following steps on the clone:
 - a. Copy `db2jcc.jar`, `db2jcc4.jar` and `db2jcc_license_cu.jar` from the `DB2SKLMV25` folder to the `<WAS_HOME>/lib` folder.

Default definition of `<WAS_HOME>` variable is typically:

Windows

`C:\Program Files (x86)\IBM\WebSphere\AppServer`

Linux `/opt/IBM/WebSphere/AppServer`

- b. Go to `<WAS_HOME>/bin`.
 - 1) Log on to `wsadmin` by using the following command:
`wsadmin.bat -user <wasadmin user> -password <wasadmin passwd>
-lang jython`
 - 2) Run the following command:
`AdminTask.deleteIdMgrDBTables<'[-schemaLocation "<WAS_HOME>/etc/wim/set
up" -databaseType db2 -dbURL "jdbc:db2://localhost:<sklmbport>/
<sklmbname>" -dbDriver com.ibm.db2.jcc.DB2Driver -dbAdminId
<sklmb2adminuser> -dbAdminPassword <sklmb2adminuserPasswd>
-reportSqlError true]'>`
- c. Follow the procedure to setup/configure LDAP integration as was done on the master IBM Security Key Lifecycle Manager server. For the integration steps, see “Integrating LDAP with IBM Security Key Lifecycle Manager” on page 164.
7. After the replication between an IBM Security Key Lifecycle Manager server that is configured for LDAP integration and a clone that is not configured for LDAP integration, if you inadvertently run the normal LDAP integration

configuration on the clone, the Step 5 in “Integrating LDAP with IBM Security Key Lifecycle Manager” on page 164 fails. You must run these steps:

a. Go to `<WAS_HOME>/bin`.

1) Log on to `wsadmin`:

```
wsadmin.bat -user <wasadmin user> -password <wasadmin passwd> -lang  
jython
```

2) Run the following command:

```
AdminTask.deleteIdMgrDBTables<'[-schemaLocation "<WAS_HOME>/etc/wim/set  
up" -databaseType db2 -dbURL "jdbc:db2://localhost:<sklmbport>/  
<sklmbname>" -dbDriver com.ibm.db2.jcc.DB2Driver -dbAdminId  
<sklmb2adminuser> -dbAdminPassword <sklmb2adminuserPasswd>  
-reportSqlError true] '>
```

b. Run steps 5 - 9 in “Integrating LDAP with IBM Security Key Lifecycle Manager” on page 164.

Exporting an SSL/KMIP server certificate

You must export the IBM Security Key Lifecycle Manager SSL/KMIP server certificate to a file in an encoded format for use by the client device. The client device imports this certificate for secure communication with the server.

About this task

Use the Export Certificate dialog, **tklmCertExport** command, or **Certificate Export REST Service** to export the IBM Security Key Lifecycle Manager SSL/KMIP server certificate to a file in an encoded format.

Procedure

1. Navigate to the appropriate page or directory:

- Graphical user interface:

Log on to the graphical user interface. The Welcome page is displayed.

- Command-line interface:

In the `WAS_HOME/bin` directory, start a **wsadmin** session by using Jython. Log on to **wsadmin** with an authorized user ID, such as the SKLMAdmin user ID. For example, on Windows systems, navigate to the `drive:\Program Files (x86)\IBM\WebSphere\AppServer\bin` directory and type:

- Windows systems:

```
wsadmin -username SKLMAdmin -password mypwd -lang jython
```

- Systems such as AIX or Linux:

```
./wsadmin.sh -username SKLMAdmin -password mypwd -lang jython
```

- REST interface:

- Open a REST client.

2. Export a certificate.

- Graphical user interface:

a. Click **Advanced Configuration > Server Certificates**.

b. In the **Certificates** table, select the appropriate certificate.

c. Click **Export**.

d. In the Export Certificate dialog, certificate that you selected in Step b is populated in the **File name** field.

e. In the **File location** field, type full path to the certificate file, or click **Browse** to specify the location.

- f. Select either **BASE64** (default format) or **DER** (Distinguished Encoding Rules) encoded file format for the certificate.

- g. Click **Export Certificate**.

- Command-line interface:

Type **tklmCertExport** to export a certificate file. For example:

```
print AdminTask.tklmCertExport
('[-uuid CERTIFICATE-61f8e7ca-62aa-47d5-a915-8adbfbdc9de
-format DER -fileName d:\\mypath\\mycertfilename.der]')
```

For more information about **tklmCertExport** command, see **tklmCertExport**.

- REST interface:

- a. Obtain a unique user authentication identifier to access IBM Security Key Lifecycle Manager REST services. For more information about the authentication process, see **Authentication process for REST services**.

- b. To start **Certificate Export REST Service**, send the HTTP PUT request. Pass the user authentication identifier that you obtained in Step a along with the request message as shown in the following example.

```
PUT https://localhost:9080/SKLM/rest/v1/certificates/export
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
{"uuid":"CERTIFICATE-61f8e7ca-62aa-47d5-a915-8adbfbdc9de",
 "format":"DER",
 "fileName":"/mycertificate.der"}
```

For more information about **Certificate Export REST Service**, see **Certificate Export REST Service**.

- 3. A success indicator varies, depending on the interface:

- Graphical user interface:

The certificate file is saved in the location you specified.

- Command-line interface:

A completion message indicates success.

- REST interface:

The status code 200 OK indicates success.

Copying a certificate between IBM Security Key Lifecycle Manager servers

You can use the command-line interface or REST interface to copy a certificate between IBM Security Key Lifecycle Manager servers with both the public and private key.

About this task

Use the following CLI commands or REST interfaces to copy a certificate:

- **tklmKeyExport** and **tklmKeyImport**
- **Key Export REST Service** and **Key Import REST Service**

Procedure

1. On the IBM Security Key Lifecycle Manager server where the certificate is located, run the **tklmKeyExport** command or send **Key Export REST Service** HTTP request.

```
print AdminTask.tklmKeyExport ('[-alias sklmCertificate
    -fileName myprivatekeys -keyStoreName defaultKeyStore
    -type privatekey -password mypassword]')
PUT https://localhost:9080/SKLM/rest/v1/keys/export
Content-Type: application/json
Accept : application/json
Authorization: SKLMAuth authId=139aeh34567m
{"alias":"sklmCertificate","fileName":"myprivatekeys","type":"privatekey",
"password":"mypassword"}
```

2. Copy the mycert.p12 file to the destination IBM Security Key Lifecycle Manager server.
3. Run the **tklmKeyImport** command or send **Key Import REST Service** HTTP request.

```
print AdminTask.tklmKeyImport ('-type privatekey -fileName c:\\mycert.p12
    -keyStoreName "Tivoli Key Lifecycle Manager Keystore" -usage 3592 -password
    <password>]')
POST https://localhost:9080/SKLM/rest/v1/keys/import
Content-Type: application/json
Accept : application/json
Authorization: SKLMAuth authId=139aeh34567m
{"type":"privatekey","fileName":"mycert.p12","usage":"3592","password":
"mypassword","newAlias":"mykey"}
```

Results

These commands copy both the private and public key to write and read tapes by using the certificate.

Changing the language of the browser interface

You can change the language that is displayed on the browser interface.

About this task

Change the language preference for your browser before you log on to IBM Security Key Lifecycle Manager. To change the language preference for your browser, complete these steps:

- Internet Explorer
 1. Select **Tools > Internet Options**.
 2. On the **General** tab, click **Languages**.
 3. Select a language and click **OK**. You might need to first add a language and move it up to the top of the list of languages.
 4. Restart the browser.
- Firefox
 1. Select **Tools > Options**. Then, click the Content icon.
 2. On the Content tab, in the Languages section, click **Choose**.
 3. Select a language and click **OK**. You might need to first add a language and move it up to the top of the list of languages.
 4. On the Options dialog, click **OK** again.
 5. Restart the browser.

Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785 U.S.A.

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law :

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement might not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
2Z4A/101
11400 Burnet Road
Austin, TX 78758 U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to

IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_. All rights reserved.

If you are viewing this information in softcopy form, the photographs and color illustrations might not be displayed.

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions.

Applicability

These terms and conditions are in addition to any terms of use for the IBM website.

Personal use

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial use

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR

IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at <http://www.ibm.com/legal/copytrade.shtml>.

Adobe, Acrobat, PostScript and all Adobe-based trademarks are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, other countries, or both.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.

UNIX is a registered trademark of The Open Group in the United States and other countries.



Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

Linear Tape-Open, LTO, the LTO Logo, Ultrium, and the Ultrium logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and other countries.

Index

Numerics

3592 153
3592 tape drive
 device.AutoPendingAutoDiscovery
 attribute 69
 tklmConfigUpdateEntry
 command 69
 tklmDeviceAdd command 69, 80
 tklmDeviceDelete command 83
 tklmDeviceList command 82
 tklmDeviceUpdate command 82

A

add
 audit records 169
 federated repository 165
 IBM Security Key Lifecycle Manager
 groups 169
 LDAP repository 165
adding, objects 157
administer
 backup and restore
 key loss prevention 153
 certificates 71, 111
 device associations 101
 devices 71, 101
 image certificates 89
 keys 101, 111
 storage images 89
administering
 audit 3
 backup and restore 115
 certificate 8
 database 35
 debug 7
 device groups 33
 groups, limiting 22, 23, 26
 groups, users, and roles 21
 KMIP certificate 1
 port 11
 role, new device group 34
 ssl certificate 1
 tasks, validating 28
administrator
 password policy, changing 30
 password, changing 31
audit
 Audit.event.outcome property 3
 Audit.event.types property 3
 level 3
 tklmConfigGetEntry command 3
 tklmConfigUpdateEntry command 3
audit records
 syslog format 5

B

backup
 automatic 19

backup (*continued*)
 Encryption Key Manager 124
 policy files 119
 unlimited strength jurisdiction policy
 files 119
 version 1.0 130
 version 2.0 136
 version 2.0.1 142
 version 2.5 148
backup and restore
 backup file, deleting 122
 jar file 116
 replica computer 115
 runtime requirements
 backup task 115
 restore task 115
 script 154
 tklm.backup.dir property 115
 tklm.db2.backup.dir property 115
 tklmBackupGetProgress
 command 122
 tklmBackupGetRestoreProgress
 command 122
 tklmBackupGetRestoreResult
 command 122
 tklmBackupGetResult command 122
 tklmBackupIsRestoreRunning
 command 122
 tklmBackupList command 122
 tklmBackupRun command 116
 tklmBackupRunRestore
 command 118
backup task
 database accessible 115
 IBM Security Key Lifecycle Manager
 running 115
browser, locale settings 174

C

cert.valiDATE
 administering 8
 certificate 8
certificate
 copy 173
 default 8
 export 172
 Get Single Config Property REST
 Service 8
 KMIP 1
 rollover 76
 ssl 1
 tklmCertCreate command 67, 73
 tklmCertDelete command 79
 tklmCertGenRequest command 1
 tklmCertImport command 73
 tklmCertUpdate command 77
 tklmConfigGetEntry command 8
 tklmConfigUpdateEntry command 8
 tklmKeyExport command 173

certificate (*continued*)
 Update Config Property REST
 Service 8
 useSKIDefaultLabels property 8
certificate export 172
 Certificate Export REST Service 172
 tklmCertExport command 172
Certificate Export REST Service,
 certificate export 172
Certificate Generate Request REST
 Service
 certificate 1
certificate request
 tklmCertGenRequest command 73,
 85, 91
 tklmCertUpdate command 77, 93
certificate, add
 IBM Spectrum Scale 112
certificate, create
 guided steps 67
certificate, delete
 IBM Spectrum Scale 113
certificate, modify
 IBM Spectrum Scale 112
change
 password policy 30
client device
 adding, objects 156, 157
client device, certificate export 156
client device, pending certificate 156
clone
 replication 14
configuration
 HSM parameters 161
 settings 1
configure
 backup script 154
create
 device groups 33
Create Certificate REST Service
 certificate 1
creating, objects
 client device 157
cross-platform
 backup and restore 124

D

data source
 database-based repository 166
DB2
 host name 41
 passwords 36, 38
 security 36, 38
 server, stopping 40
 transactional logs, maintaining 35
debug
 debug property 7
 tklmConfigGetEntry command 7
 tklmConfigUpdateEntry command 7
deleting, client device 159

- deleting, objects 159
- device
 - moving between groups 44
 - pending 41
- device group, move to another 44
- device.AutoPendingAutoDiscovery
 - 3592 tape drive 69
 - LTO tape drive 49
- device.AutoPendingAutoDiscovery
 - DS8000 87
- DS5000 storage server
 - tklmDeviceAdd command 102
 - tklmDeviceDelete command 105
 - tklmDeviceList command 104
 - tklmDeviceUpdate command 104
- DS8000 Turbo drive
 - device.AutoPendingAutoDiscovery
 - attribute 87
 - tklmDeviceAdd command 97
 - tklmDeviceDelete command 99
 - tklmDeviceGroupAttributeUpdate
 - command 87
 - tklmDeviceList command 98
 - tklmDeviceUpdate command 98

E

Encryption Key Manager 124

F

federated repository 165

G

- Get Single Config Property REST Service
 - certificate 8
 - port 11
- global security
 - disable 121
 - enable 121
- guided steps
 - certificate, create 67
 - key group, create 47
 - storage image, create 85

H

- Hardware Security Module
 - configuration 159
 - configuration requirements 162
 - pkcs11.config 159
 - pkcs11.pin 159
 - pkcs11.pin.obfuscated 159
- host name
 - DB2 server 41
 - WebSphere Application Server 41
- HSM
 - configuration 159
 - configuration requirements 162
 - IBM 4765 PCIe Cryptographic
 - Coprocessor 159
 - nCipher nShield Connect 159
 - SafeNet Luna SA 159

- HSM parameters
 - configuration 161
 - pkcs11.config 161
 - pkcs11.pin 161
 - pkcs11.pin.obfuscated 161

I

- IBM Security Key Lifecycle Manager user
 - password, changing 32
- IBM Spectrum Scale
 - certificate, add 112
 - certificate, delete 113
 - certificate, modify 112
 - key, add 113
 - key, delete 114
 - key, modify 114
- image certificate
 - tklmCertCreate command 85, 91
 - tklmCertDelete command 95
 - tklmCertImport command 91
 - tklmCertUpdate command 93
- importing, client certificate 156
- installation
 - DB2
 - password 36, 38
 - security 36, 38
 - host name
 - DB2 server 41
 - WebSphere Application Server 41

J

- jar file, backup and restore 116
- JCE unlimited strength jurisdiction 119

K

- key
 - tklmGroupCreate command 107
 - tklmGroupEntryAdd command 109
 - tklmGroupEntryDelete
 - command 109
 - tklmGroupList command 107
 - tklmKeyDelete command 60
 - tklmKeyList command 60
 - tklmSecretKeyCreate command 107
- key group
 - rollover 55
 - stopRoundRobinKeyGrps
 - property 57
 - tklmGroupCreate command 47, 54
 - tklmGroupDelete command 60
 - tklmGroupEntryAdd command 58
 - tklmGroupEntryDelete command 58
 - tklmGroupList command 47, 54
 - tklmSecretKeyCreate command 54
- key group, create
 - guided steps 47
- key, add
 - IBM Spectrum Scale 113
- key, delete
 - IBM Spectrum Scale 114
- key, modify
 - IBM Spectrum Scale 114
- klmAdminDeviceGroup permission 23

- klmAudit permission 23
- klmBackup permission 23
- klmConfigure permission 23
- klmCreate permission 23
- klmDelete permission 23
- klmGet permission 23
- klmModify permission 23
- klmRestore permission 23
- klmView permission 23

KMIP

- attributes 156
- objects 156
- operations 156
- KMIP objects 156

L

- language, preference 174
- LDAP integration
 - IBM Security Key Lifecycle
 - Manager 162, 164, 170
 - user repositories
 - LDAP 162, 164, 170
 - LDAP repository 165, 167
 - LDAP users
 - IBM Security Key Lifecycle Manager
 - groups 169
- locale, browser settings 174
- LTO 153
- LTO tape drive
 - device.AutoPendingAutoDiscovery
 - attribute 49
 - symmetricKeySet attribute 49
 - tklmDeviceAdd command 49, 62
 - tklmDeviceDelete command 65
 - tklmDeviceGroupAttributeUpdate
 - command 49
 - tklmDeviceList command 64
 - tklmDeviceUpdate command 64

M

- manage
 - 3592 tape drive 67
 - devices 51
 - DS5000 storage server 101
 - DS8000 Turbo drive 85
 - IBM Spectrum Scale 110
 - key groups 51
 - keys 51
 - LTO tape drive 47
- master
 - replication 17
- modifying, client device 158
- modifying, objects 158

P

- password
 - DB2 36, 38
 - policy 29
 - strength 29
- password change
 - IBM Security Key Lifecycle Manager
 - user 32
- pending device 41

- permissions
 - klmAdminDeviceGroup 23
 - klmAudit 23
 - klmBackup 23
 - klmConfigure 23
 - klmCreate 23
 - klmDelete 23
 - klmGet 23
 - klmModify 23
 - klmRestore 23
 - klmView 23
- policy files, backup 119
- port
 - default 11
 - Get Single Config Property REST Service 11
 - number
 - conflicts 11
 - current value 11
 - determining current 11
 - KMIP SSL 11
 - SSL 11
 - TCP 11
 - ssl 11
 - tcp 11
 - timeout 11
 - tklmConfigGetEntry command 11
 - tklmConfigUpdateEntry
 - command 11
 - TransportListener.ssl.port property 11
 - TransportListener.ssl.timeout
 - property 11
 - TransportListener.tcp.port
 - property 11
 - TransportListener.tcp.timeout
 - property 11
- post-installation steps
 - DB2
 - transactional logs, maintaining 35
 - DB2, stop 40
 - WebSphere Application Server 41

R

- register
 - client device 156
- registering, client device 156
- replication
 - clone 13, 17
 - master 13, 14, 19
- repository, database-based
 - application groups 167
 - data source 166
- restore
 - version 1.0 132
 - version 2.0 138
 - version 2.0.1 144
 - version 2.1 126
 - version 2.5 150
- restore task
 - database accessible 115
 - password requirement 115
 - primary computer 115
- role
 - group 168
 - user 168

- role, administrator
 - klmGUICLIAccessGroup 168
- roles
 - assignment to group 23
 - suppressmonitor 23
- rollover
 - certificate 76
 - certificates 153
 - key group 55
 - key groups 153

S

- script
 - backup 154
- security
 - DB2 36, 38
- session
 - wsadmin, using Jython 31, 32
- settings
 - configuration 1
- startServer
 - command 120
 - script 120
- stopRoundRobinKeyGrps, property 57
- stopServer
 - command password, caution displaying 120
 - global security user ID, password 120
 - script 120
- storage image
 - tklmDeviceAdd command 97
 - tklmDeviceDelete command 99
- storage image, create
 - guided steps 85
- strength, password 29
- suppressmonitor role 23
- symmetric key
 - key group 47
 - tklmSecretKeyCreate command 47
- syslog format
 - audit records 5

T

- tklm.backup.dir, backup and restore 115
- tklm.db2.backup.dir, backup and restore 115
- tklmBackupGetProgress, backup and restore 122
- tklmBackupGetRestoreProgress, backup and restore 122
- tklmBackupGetRestoreResult, backup and restore 122
- tklmBackupGetResult, backup and restore 122
- tklmBackupIsRestoreRunning, backup and restore 122
- tklmBackupList, backup and restore 122
- tklmBackupRun, backup and restore 116
- tklmBackupRunRestore, backup and restore 118
- tklmCertCreate
 - certificate 1, 67, 73
 - image certificate 85, 91

- tklmCertDelete
 - certificate 79
 - image certificate 95
- tklmCertExport command, certificate export 172
- tklmCertGenRequest
 - certificate request 67, 73, 85, 91
- tklmCertImport
 - certificate 73
 - image certificate 91
- tklmCertUpdate
 - certificate 77
 - certificate request 77, 93
 - image certificate 93
- tklmConfigGetEntry
 - audit 3
 - certificate 8
 - debug 7
 - port 11
- tklmConfigUpdateEntry
 - 3592 tape drive 69
 - audit 3
 - certificate 8
 - debug 7
- tklmDeviceAdd
 - 3592 tape drive 69, 80
 - DS5000 storage server 102
 - DS8000 Turbo drive 87, 97
 - LTO tape drive 49, 62
 - storage image 97
- tklmDeviceDelete
 - 3592 tape drive 83
 - DS5000 storage server 105
 - DS8000 Turbo drive 99
 - LTO tape drive 65
 - storage image 99
- tklmDeviceGroupAttributeUpdate
 - DS8000 Turbo drive 87
 - LTO tape drive 49
- tklmDeviceList
 - 3592 tape drive 82
 - DS8000 Turbo drive 98
 - LTO tape drive 64, 104
- tklmDeviceUpdate
 - 3592 tape drive 82
 - DS5000 storage server 104
 - DS8000 Turbo drive 98
 - LTO tape drive 64
- tklmGroupCreate
 - key 107
 - key group 47, 54
- tklmGroupDelete, key group 60
- tklmGroupEntryAdd
 - key 109
 - key group 58
- tklmGroupEntryDelete
 - key 109
 - key group 58
- tklmGroupList
 - key 107
 - key group 47, 54
- tklmKeyDelete, key 60
- tklmKeyExport, copy certificate 173
- tklmKeyImport command 173
- tklmKeyImport, copy certificate 173
- tklmKeyList, key 60

- tklmSecretKeyCreate
 - key 107
 - key group 54
 - symmetric key 47
- TransportListener.ssl.port,
 - administering 11
- TransportListener.ssl.timeout,
 - administering 11
- TransportListener.tcp.port,
 - administering 11
- TransportListener.tcp.timeout,
 - administering 11

U

- Update Config Property REST Service
 - certificate 8
 - port 11
- useSKIDefaultLabels
 - administering 8
 - certificate 8

V

- version 1.0, backup 130
- version 1.0, restore 132
- version 2.0, backup 136

- version 2.0, restore 138
- version 2.0.1, backup 142
- version 2.0.1, restore 144
- version 2.1, restore 126
- version 2.5, backup 148
- version 2.5, restore 150
- version, earlier 124

W

- WebSphere Application Server
 - host name, changing 41