Administering

Contents

Administering

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

The administrative activities include the following tasks:

- Setting up and maintaining IBM Security Key Lifecycle Manager system
- Setting up the master and clone systems for replication
- Administering the groups, users, and roles
- Administering devices, KMIP objects, and Hardware Security Module
- Running operational tasks such as data backup, data restore, and export/import of device groups
- Other miscellaneous administrative tasks

Before you begin, familiarize yourself with the concepts and terminologies that are mentioned in this section. See the Overview, Planning, and Installing and configuring sections for the related information.

Specifying SSL or KMIP certificates

You can specify the self-signed certificate to be used as server communication certificate. Alternatively, you can 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 `C:\Program Files\IBM\WebSphere\AppServer\products\sklm\171029122037—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. Go 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
     a. Go to the *<WAS_HOME>*/bin directory. For example,

        Windows
        ```
        cd drive:\Program Files\IBM\WebSphere\AppServer\bin
        ```

        Linux
        ```
        cd /opt/IBM/WebSphere/AppServer/bin
        ```
     b. Start the **wsadmin** interface by using an authorized user ID, such as SKLMAdmin. For example,

        Windows
        ```
        wsadmin.bat -username SKLMAdmin -password mypwd -lang jython
        ```

        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 run **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:<port>/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:<port>/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"}
```

If you select a certificate request for a third-party provider, the certificate request file in .csr format is generated in the <SKLM_HOME> directory, for example, C:\Program Files\IBM\WebSphere\AppServer\products\sklm\171029122037–sslcert001.csr. Manually send the certificate request to a certificate authority. You must then import the signed certificate to IBM Security Key Lifecycle Manager. For the steps about how to send and import the certificate, see Scenario: Request for a third-party certificate.

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

Depending on your need, you can 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 the audit information levels (Low, Medium, or High) 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. Go 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
  a. Go to the *<WAS_HOME>*/bin directory. For example,

     Windows

     ```
     cd drive:\Program Files\IBM\WebSphere\AppServer\bin
     ```

     Linux

     ```
     cd /opt/IBM/WebSphere/AppServer/bin
     ```

  b. Start the **wsadmin** interface by using an authorized user ID, such as SKLMAdmin. For example,

     Windows

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

     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 run **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:<port>/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:<port>/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. Restart the server. For instructions about how to stop and start the server, see "Restarting the IBM Security Key Lifecycle Manager server" on page 190.

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 use the IBM Security Key Lifecycle Manager graphical user interface to configure and generate the audit records in syslog format and send them to a syslog server.

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. If you need the secure transfer of audit information to the syslog server by using the SSL/TLS transport protocol, select **Use SSL/TLS**.
7. Click **OK**.

What to do next

After you enabled syslog format for audit records with the necessary 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:<port>/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 the server. For instructions about how to stop and start the server, see "Restarting the IBM Security Key Lifecycle Manager server" on page 190.

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. Go 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
     a. Go to the `<WAS_HOME>/bin` directory. For example,

        Windows
        ```
        cd drive:\Program Files\IBM\WebSphere\AppServer\bin
        ```
        Linux
        ```
        cd /opt/IBM/WebSphere/AppServer/bin
        ```
     b. Start the **wsadmin** interface by using an authorized user ID, such as `SKLMAdmin`. For example,

        Windows
        ```
        wsadmin.bat -username SKLMAdmin -password mypwd -lang jython
        ```
        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 run **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:<port>/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:<port>/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:

Specifying port and timeout settings

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

About this task

You can use the Key Serving Ports 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. Go 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
     a. Go to the *<WAS_HOME>*/bin directory. For example,

        Windows
        ```
        cd drive:\Program Files\IBM\WebSphere\AppServer\bin
        ```
        Linux
        ```
        cd /opt/IBM/WebSphere/AppServer/bin
        ```
     b. Start the **wsadmin** interface by using an authorized user ID, such as SKLMAdmin. For example,

        Windows
        ```
        wsadmin.bat -username SKLMAdmin -password mypwd -lang jython
        ```
        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.

     IBM Security Key Lifecycle Manager agent port
     Agent uses default port 60015 to communicate with IBM Security

Key Lifecycle Manager server. You can update the default agent port number only when the IBM Security Key Lifecycle Manager instance is not configured for multi-master setup.

- 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 run **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:<port>/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:<port>/SKLM/rest/v1/configProperties
     Content-Type: application/json
     Accept: application/json
     Authorization: SKLMAuth authId=139aeh34567m
     Accept-Language: en
     {"TransportListener.tcp.port": "3802"}
     ```

What to do next

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

Checking the current port number

After IBM Security Key Lifecycle Manager server installation, you might want to determine the secure and non-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_adminhost_secure**, **WC_defaulthost**, and the **WC_defaulthost_secure**, property in the *WAS_HOME*/profiles/KLMProfile/properties/portdef.props file. For example, the file might specify these values:

```
WC_adminhost_secure=9083
WC_defaulthost=80
WC_defaulthost_secure=443
```

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

Specifying key serving parameters

You can 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. Go 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
     a. Go to the <*WAS_HOME*>/bin directory. For example,

        Windows
        ```
        cd drive:\Program Files\IBM\WebSphere\AppServer\bin
        ```

        Linux
        ```
        cd /opt/IBM/WebSphere/AppServer/bin
        ```

     b. Start the **wsadmin** interface by using an authorized user ID, such as SKLMAdmin. For example,

        Windows
        ```
        wsadmin.bat -username SKLMAdmin -password mypwd -lang jython
        ```

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 run **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:<port>/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:<port>/SKLM/rest/v1/configProperties
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth authId=139aeh34567m
Accept-Language: en
{ "zOSCompatibility": "true"}
```

What to do next

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

Truststore configuration

The IBM Security Key Lifecycle Manager truststore stores the trusted certificates and the device root certificates that are used for secure communication between IBM Security Key Lifecycle Manager and the devices.

The installation of IBM Security Key Lifecycle Manager creates the truststore file `tklmTruststore.jceks` at `<WAS_HOME>\products\sklm\keystore`.

Windows
`C:\Program Files\IBM\WebSphere\AppServer\products\sklm\keystore`

Linux `/opt/IBM/WebSphere/AppServer/products/sklm/keystore`

You can add the device root certificate to the trust list by adding it to the truststore. When the device root certificate is added to the truststore, all devices with certificates that are signed by this root certificate are automatically becomes trusted. Adding device root certificate eliminates the need to import device certificate to the client device communication certificate list to establish SSL/TLS communication with IBM Security Key Lifecycle Manager server.

You can use IBM Security Key Lifecycle Manager graphical user interface, command line interface, and REST interface to manage certificates in the truststore.

You can run the following actions on the certificates:

- Add a certificate to the truststore.
- View the certificates in the truststore.
- Delete certificates from the truststore.

Adding a certificate to the truststore

You might add a certificate from a certificate file that is in DER or base64 format to the IBM Security Key Lifecycle Manager internal truststore. The certificate is used for communication between IBM Security Key Lifecycle Manager and the device that identifies itself by using this certificate or the root certificate for this certificate.

About this task

You can use the Add Certificate dialog, **tklmTrustStoreCertAdd** command, or **Truststore Certificate Add REST Service** to add a certificate to the IBM Security Key Lifecycle Manager truststore. Your user ID must have the klmSecurityOfficer role.

Procedure

1. Go 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** > **Truststore**.
     c. On the Truststore page, click **Add**.
   - Command-line interface
     a. Go to the <*WAS_HOME*>/bin directory. For example,

        Windows
        ```
        cd drive:\Program Files\IBM\WebSphere\AppServer\bin
        ```
        Linux
        ```
        cd /opt/IBM/WebSphere/AppServer/bin
        ```
     b. Start the **wsadmin** interface by using an authorized user ID, such as SKLMAdmin. For example,

        Windows
        ```
        wsadmin.bat -username SKLMAdmin -password mypwd -lang jython
        ```
        Linux
        ```
        ./wsadmin.sh -username SKLMAdmin -password mypwd -lang jython
        ```
   - REST interface
     - Open a REST client.
2. Add a certificate from a certificate file that is in DER or base64 format to the truststore.
   - Graphical user interface
     a. In the **Certificate alias** field, specify alias name for the certificate.
     b. Click **Browse** to specify the certificate file location under <*SKLM_DATA*> directory, for example, C:\Program Files\IBM\WebSphere\AppServer\products\sklm\data. For the definition of <*SKLM_DATA*>, see Definitions for *HOME* and other directory variables.
     c. Select the certificate file format such as DER or base64.
     d. Click **Add Certificatre**.
   - Command-line interface

     Type **tklmTrustStoreCertAdd** to add a certificate file to the truststore. For example, to add a certificate file in DER format, run the following command.
     ```
     print AdminTask.tklmTrustStoreCertAdd
      ('[-fileName d:\\mypath\\mycertfilename.der
       -format DER -alias myCertAlias]')
     ```

- REST interface

  Use **Truststore Certificate Add REST Service** to add a certificate. For example, you can send the following HTTP request.

  ```
  PUT https://localhost:<port>/SKLM/rest/v2/trustStoreCertificates/addCertToTrustStore
  Content-Type: application/json
  Accept: application/json
  Authorization: SKLMAuth userAuthId=139aeh34567m
  {"certFile":"C:\\Program Files\\IBM\\WebSphere\\AppServer\\products\\sklm\\data\\clientsslce
  "certFormat":"DER","certAlias":"myCert"}
  ```

Deleting a certificate from the truststore

You might delete a certificate that is in the IBM Security Key Lifecycle Manager internal truststore. For example, you might delete a certificate for which a business needs no longer exists.

About this task

You can use the Delete dialog, **tklmTrustStoreCertDelete** command, or **Truststore Certificate Delete REST Service** to delete a certificate from the truststore. Your user ID must have the `klmSecurityOfficer` role.

Procedure

1. Go 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** > **Truststore**.
     c. On the Truststore page, select a certificate.
     d. Click **Delete**.
     e. Alternatively, right-click a certificate on the table and then select **Delete**.
   - Command-line interface
     a. Go to the *<WAS_HOME>*/bin directory. For example,

        Windows
        ```
        cd drive:\Program Files\IBM\WebSphere\AppServer\bin
        ```

        Linux
        ```
        cd /opt/IBM/WebSphere/AppServer/bin
        ```
     b. Start the **wsadmin** interface by using an authorized user ID, such as SKLMAdmin. For example,

        Windows
        ```
        wsadmin.bat -username SKLMAdmin -password mypwd -lang jython
        ```

        Linux
        ```
        ./wsadmin.sh -username SKLMAdmin -password mypwd -lang jython
        ```
   - REST interface
     – Open a REST client.
2. Delete the certificate from the truststore.
   - Graphical user interface

     On the Confirm dialog, read the confirmation message before you delete the certificate. Click **OK**.
   - Command-line interface

Use the tklmTrustStoreCertList command to find a certificate, and tklmTrustStoreCertDelete command to delete a certificate from the truststore. To find the certificate, run the following command.

```
print AdminTask.tklmTrustStoreCertList ('[-alias myCertAlias]')
```

To delete a certificate from the truststore, run the following command.

```
print AdminTask.tklmTrustStoreCertDelete ('[-alias myCertAlias]')
```

- REST interface

  Use Truststore Certificate List REST Service to find a certificate and Truststore Certificate Delete REST Service to delete a certificate from the truststore. For example, you can send the following HTTP requests by using a REST client.

```
GET https://localhost:<port>/SKLM/rest/v1/trustStoreCertificates?alias=myCertAlias
Content-Type: application/json
Accept : application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
```

```
DELETE https://localhost:<port>/SKLM/rest/v2/trustStoreCertificates?alias=myCertAlias
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
```

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.

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.ibm.com/support/knowledgecenter/en/SSAW57_9.0.0/com.ibm.websphere.nd.multiplatform.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

a. Go to the *<WAS_HOME>*`/bin` directory. For example,

Windows

```
cd drive:\Program Files\IBM\WebSphere\AppServer\bin
```

Linux

```
cd /opt/IBM/WebSphere/AppServer/bin
```

b. Start the **`wsadmin`** interface by using an authorized user ID, such as WASAdmin. For example,

Windows

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

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, under **Role(s)**, select the required subset of roles from the 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.

b. Select **Map Groups As Specified Below**.

c. Type a search string in the **Search string** field. For example, type `DS5000Admin`.

d. Click **Search**.

e. From the **Available** list, select the group.

f. Click the arrow to move selected group to the **Mapped to role** column.

g. Click **OK**.

h. 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.ibm.com/support/knowledgecenter/en/SSAW57_9.0.0/com.ibm.websphere.nd.multiplatform.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
     a. Go to the `<WAS_HOME>/bin` directory. For example,

        Windows

        ```
        cd drive:\Program Files\IBM\WebSphere\AppServer\bin
        ```

Linux `cd /opt/IBM/WebSphere/AppServer/bin`

b. Start the **`wsadmin`** interface by using an authorized user ID, such as WASAdmin. For example,

Windows

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

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.

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: `klmGUICLIAccessGroup`

For more information about the commands that create groups and users, see the IBM WebSphere Application Server documentation (http://www.ibm.com/support/knowledgecenter/en/SSAW57_9.0.0/com.ibm.websphere.nd.multiplatform.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
     a. Go to the `<WAS_HOME>/bin` directory. For example,

        Windows
        ```
        cd drive:\Program Files\IBM\WebSphere\AppServer\bin
        ```

        Linux
        ```
        cd /opt/IBM/WebSphere/AppServer/bin
        ```

     b. Start the **`wsadmin`** interface by using an authorized user ID, such as WASAdmin. For example,

        Windows
        ```
        wsadmin.bat -username WASAdmin -password wasadminpw -lang jython
        ```

        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.

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.ibm.com/support/knowledgecenter/en/SSAW57_9.0.0/com.ibm.websphere.nd.multiplatform.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 as shown in the following example.
     - Go to the `<`*`WAS_HOME`*`>/bin` directory.

       Windows
       ```
       cd drive:\Program Files\IBM\WebSphere\AppServer\bin
       ```

       Linux
       ```
       cd /opt/IBM/WebSphere/AppServer/bin
       ```
     - Start the **`wsadmin`** interface by using an authorized user ID.

Windows

```
wsadmin.bat -username myAdmin -password password -lang jython
```

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:<port>/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_DATA*/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
Upper-case characters	At least 1
Lower-case characters	At least 1

Table 1. Password rules (continued)

Rule	**Default value**
Special characters You can use any special characters that are available on the keyboard. The special character requirement is not enforced when **imcl** tool is used for silent installation.	At least 1
Disallow the presence of the user ID* in the password	Enabled
Disallow the presence of the user name* in the password	Enabled

* Detection of this value is case-sensitive.
Note: To specify that the value is not case-sensitive, edit the default password policy and specify `CaseInsensitive` for the user ID and user name:

```
<?xml version="1.0" encoding="UTF-8"?>
<PasswordPolicy enabled="true" name="Password policy for TKLM"
 uuid="" version="1.0">
 <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"/>
         <NotUserID/>
         <NotUserName/>
     </PasswordRuleSet>
  ]]></PasswordRules>
</PasswordPolicy>
```

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_DATA*/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.ibm.com/support/knowledgecenter/en/SSAW57_9.0.0/com.ibm.websphere.nd.multiplatform.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
     a. Go to the `<`*`WAS_HOME`*`>/bin` directory. For example,

        Windows
        ```
        cd drive:\Program Files\IBM\WebSphere\AppServer\bin
        ```

        Linux
        ```
        cd /opt/IBM/WebSphere/AppServer/bin
        ```

     b. Start the **`wsadmin`** interface by using an authorized user ID, such as WASAdmin. For example,

        Windows
        ```
        wsadmin.bat -username WASAdmin -password wasadminpw -lang jython
        ```

        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.

Resetting a password

You must be the administrator to reset a password for the IBM Security Key Lifecycle Manager or WebSphere Application Server.

About this task

You can reset the password on the computer on which IBM Security Key Lifecycle Manager runs. Use these steps only when the password of the user is lost. In all other cases, use the graphical user interface to update the password.

Procedure

1. Log in with the a local administrator user ID.
2. Back up the `WAS_HOME/profiles/KLMProfile/config/cells/SKLMCell/fileRegistry.xml` file. Changing the value of the password changes this registry file.
3. Change the password.
   - Windows
     a. Start a **wsadmin** session by using the Jython syntax. For example, type:
     ```
     WAS_HOME/bin/wsadmin.bat -conntype none -profileName KLMProfile -lang jython
     ```
     b. Reset the password for the SKLMAdmin user ID:
     ```
     wsadmin>print AdminTask.changeFileRegistryAccountPassword
      ('-userId SKLMAdmin -password newpassword')
     ```

     Note:
     - Only the WASAdmin user ID or another user ID with WebSphere Application Server administrator authority can change passwords by using the **AdminTask.changeFileRegistryAccountPassword** command.
     - Passwords that you create by using the **AdminTask.changeFileRegistryAccountPassword** command are not validated against the configured password policy that IBM Security Key Lifecycle Manager provides.

       After a lost password reset, the user must set the password by using the graphical user interface.

c. Save the change and exit:

```
wsadmin>print AdminConfig.save()
wsadmin>exit
```

- Linux

  a. Start a **wsadmin** session by using the Jython syntax. For example, type on one line:

  ```
  WAS_HOME/bin/wsadmin.sh -conntype none
     -profileName KLMProfile -lang jython
  ```

  b. Reset the password for the SKLMAdmin user ID:

  ```
  wsadmin>print AdminTask.changeFileRegistryAccountPassword
   ('-userId SKLMAdmin -password newpassword')
  ```

  Note:

  – Only the WASAdmin user ID or another user ID with IBM Security Key Lifecycle Manager administrator authority can change passwords by using the **AdminTask.changeFileRegistryAccountPassword** command.

  – Passwords that you create by using the **AdminTask.changeFileRegistryAccountPassword** command are not validated against the configured password policy that IBM Security Key Lifecycle Manager provides.

  After a lost password reset, the user must set the password by using the graphical user interface.

  c. Save the change and exit:

  ```
  wsadmin>print AdminConfig.save()
  wsadmin>exit
  ```

4. Stop and start the server.

   - Stop

     Windows

     ```
     stopServer.bat server1
     ```

     Linux

     ```
     ./stopServer.sh server1
     ```

   - Start

     Windows

     ```
     startServer.bat server1
     ```

     Linux

     ```
     ./startServer.sh server1
     ```

5. Verify that you can log in as the specified administrator with the new password.

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.ibm.com/support/knowledgecenter/SSAW57_9.0.0/com.ibm.websphere.nd.multiplatform.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\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:

       ```
       AdminTask.changeMyPassword('[-oldPassword <oldpasswordvalue>
       -newPassword
       <newpasswordvalue> -confirmNewPassword <newpasswordvalue>]')
       ```

       Example:

       ```
       AdminTask.changeMyPassword('[-oldPassword sklmadmin -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 (**klmAdminDeviceGroup**)

If you have the **klmAdminDeviceGroup** 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 **klmAdminDeviceGroup** 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
     a. Go to the `<WAS_HOME>/bin` directory. For example,

        Windows
        ```
        cd drive:\Program Files\IBM\WebSphere\AppServer\bin
        ```
        Linux
        ```
        cd /opt/IBM/WebSphere/AppServer/bin
        ```
     b. Start the **wsadmin** interface by using an authorized user ID, such as `SKLMAdmin`. For example,

        Windows
        ```
        wsadmin.bat -username SKLMAdmin -password mypwd -lang jython
        ```
        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:<port>/SKLM/rest/v1/deviceGroups/newGroup
        Content-Type: application/json
        Accept : application/json
        Authorization: SKLMAuth authId=139aeh34567m
        {"deviceFamily":"LTO","shortName":"myLTO","longName":"my companyname
        LTO 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 ('[-name myLTO -v y]')
  ```

- REST interface:

  Send the following HTTP GET request by using a REST client:

  ```
  GET https://localhost:<port>/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/SKLMCell/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 “Restarting the IBM Security Key Lifecycle Manager server” on page 190.

What to do next

Next, you can 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:\sklmdbarchive\sklmdb31\SKLMDB301\NODE0000\LOGSTREAM0000\C0000000
```

where:

- *INSTANCEHOME* is the drive letter that you specified during the installation.
- `sklmdb31` is the database instance owner.
- `sklmdb31` 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:

```
~sklmdbarchive/sklmdb301/sklmdb31/NODE0000/LOGSTREAM0000/C0000000
```

where:

- `sklmdb31` is the database instance owner.
- `sklmdb31` 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 `sklmdbarchive` 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=sklmdb31
   db2 get db cfg for SKLMDB31
   ```

   Systems such as Linux or AIX:

   ```
   db2 get db cfg for SKLMDB31
   ```

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 `sklmdbarchive` directory to the new directory.

   Logs are named `S`*`nnnnnnn`*`.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 `S0000000.LOG`.

   Note: Running a restore operation removes the `sklmdbarchive` 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.

Run the following steps to change the Db2 database password:

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 - `DB2SKLMV301` - *sklminstance*

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

     `DB2 - DBSKLMV301 - SKLMDB31`

   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 (`DBSKLMV31`)
   - Db2 Management Service (`DBSKLMV31`)
   - Db2 Governor (`DBSKLMV31`)
   - DB Remote Command Server (`DBSKLMV31`)
6. Start the WebSphere Application Server.
7. Using the **`wsadmin`** interface that the WebSphere Application Server provides, specify the Jython syntax.

   Windows

   ```
   wsadmin.bat -username WASAdmin -password mypwd -lang jython
   ```

Linux

```
./wsadmin.sh -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 passw0rd]]')
   ```

   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)','[[passwo
   ```

   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 V9.0 - SKLM301Server.

   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 Linux or AIX systems

On Linux or AIX systems, 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 sklmdb31. Additionally, the Db2 application creates an operating system user entry named sklmdb31. 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 user at the operating system level.

1. Log on to IBM Security Key Lifecycle Manager server as root.
2. Change user to the sklmdb31 system user entry. Type:

   ```
   su sklmdb31
   ```

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 sklmdb31}
      ```

      And also type on one line:

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

The result is like this example:

```
{alias sklmdb}
{description "SKLM database user J2C authentication alias"}
{password *****}
{userId sklmdb31}
```

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 sklmdb 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/KLMProfile/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 AIX or Linux systems, or the Local Administrator on Windows systems.

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\IBM\WebSphere\AppServer\bin
   .\stopServer.bat server1 -username wasadmin -password mysecretpwd
   ```

   AIX or Linux systems
   ```
   /opt/IBM/WebSphere/AppServer/bin/stopServer.sh server1 -username wasadmin
     -password mysecretpwd
   ```
3. Stop the Db2 server. Type these commands.

   Windows systems
   ```
   set DB2INSTANCE=sklmdb31
   db2stop
   ```

   AIX or Linux systems
   ```
   su -sklmdb31
   db2stop
   ```

Changing the Db2 server host name

After you change the IBM Security Key Lifecycle Manager system host name, you must 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.ibm.com/support/knowledgecenter/SSEQTP_9.0.0/com.ibm.websphere.base.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" on page 37.

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 and 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 key group

As a first activity, create keys and key groups for IBM Security Key Lifecycle Manager. Before you begin, determine the quantity of keys and the purpose of individual key groups that your organization requires.

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.

Procedure

1. Go 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

a. Go to the *<WAS_HOME>*/bin directory. For example,

Windows

```
cd drive:\Program Files\IBM\WebSphere\AppServer\bin
```

Linux

```
cd /opt/IBM/WebSphere/AppServer/bin
```

b. Start the **wsadmin** interface by using an authorized user ID, such as SKLMAdmin. For example,

Windows

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

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 run **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:<port>/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:<port>/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:<port>/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"}
```

What to do next

Back up new keys before the keys are served to devices. You can 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

Identify an LTO tape drive for use with IBM Security Key Lifecycle Manager. Before you begin, create the key groups that you want to associate with tape drives that you identify.

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.

You can make any of the following choices for serving keys to devices.

Only accept manually added devices for communication
All incoming devices are not added to the data store. You must manually specify key service to each device.

Hold new device requests pending my approval
All incoming devices of a valid device group are added to the device store, 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.

Automatically accept all new device requests for communication
All new incoming devices of a valid device group are added to the data store and are automatically served keys upon request.

Note: Do not use this setting if you intend to move the new device to another device group. Instead, select manual or pending approval mode to allow an opportunity to move the device into the appropriate device group before any keys are served.

Any setting is acceptable if there are no device groups. However, if device groups are specified:

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. Go 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
     a. Go to the *<WAS_HOME>*/bin directory. For example,

        Windows

        ```
        cd drive:\Program Files\IBM\WebSphere\AppServer\bin
        ```

        Linux

        ```
        cd /opt/IBM/WebSphere/AppServer/bin
        ```

     b. Start the **wsadmin** interface by using an authorized user ID, such as SKLMAdmin. For example,

        Windows

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

        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 run **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:<port>/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:<port>/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"}
     ```

What to do next

Next, you can 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 can 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 can 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 can do the following 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 can 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 can add more keys or key groups for use with IBM Security Key Lifecycle Manager. Before you begin, determine your site policy on the default key groups and naming for key prefixes.

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.

Procedure

1. Go 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
     a. Go to the *<WAS_HOME>*/bin directory. For example,

        Windows
        ```
        cd drive:\Program Files\IBM\WebSphere\AppServer\bin
        ```

        Linux
        ```
        cd /opt/IBM/WebSphere/AppServer/bin
        ```
     b. Start the **wsadmin** interface by using an authorized user ID, such as SKLMAdmin. For example,

        Windows
        ```
        wsadmin.bat -username SKLMAdmin -password mypwd -lang jython
        ```

        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:<port>/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:<port>/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:<port>/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"}
     ```

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 can 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. Go 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
  a. Go to the *<WAS_HOME>*/bin directory. For example,

     Windows
     ```
     cd drive:\Program Files\IBM\WebSphere\AppServer\bin
     ```

     Linux
     ```
     cd /opt/IBM/WebSphere/AppServer/bin
     ```

  b. Start the **wsadmin** interface by using an authorized user ID, such as SKLMAdmin. For example,

     Windows
     ```
     wsadmin.bat -username SKLMAdmin -password mypwd -lang jython
     ```

     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 LTO]')
     ```

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

Specifying that keys are used only once

You can specify that the 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 or REST interface to set 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.

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. Go to the appropriate directory:
   a. Go to the <*WAS_HOME*>/bin directory. For example,

      Windows
      ```
      cd drive:\Program Files\IBM\WebSphere\AppServer\bin
      ```

      Linux
      ```
      cd /opt/IBM/WebSphere/AppServer/bin
      ```

   b. Start the **wsadmin** interface by using an authorized user ID, such as SKLMAdmin. For example,

      Windows
      ```
      wsadmin.bat -username SKLMAdmin -password mypwd -lang jython
      ```

      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:<port>/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:<port>/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 can modify information about objects in a key group in the IBM Security Key Lifecycle Manager database. 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.

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.

Procedure

1. Go 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

a. Go to the <*WAS_HOME*>/bin directory. For example,

Windows

```
cd drive:\Program Files\IBM\WebSphere\AppServer\bin
```

Linux

```
cd /opt/IBM/WebSphere/AppServer/bin
```

b. Start the **wsadmin** interface by using an authorized user ID, such as SKLMAdmin. For example,

Windows

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

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 aaa000000000000000000}
   {keyStoreName defaultKeyStore}"]')
```

- REST interface:

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

```
DELETE https://localhost:<port>/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:<port>/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 can use the LTO Key and Device Management page to associate the key group with specific devices.

Deleting a key or key group

You can 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. Go 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
     a. Go to the `<*WAS_HOME*>/bin` directory. For example,

        Windows

        ```
        cd drive:\Program Files\IBM\WebSphere\AppServer\bin
        ```

        Linux

        ```
        cd /opt/IBM/WebSphere/AppServer/bin
        ```

     b. Start the **`wsadmin`** interface by using an authorized user ID, such as `SKLMAdmin`. For example,

        Windows

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

        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 aaa000000000000000000
        -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:<port>/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:<port>/SKLM/rest/v1/keys/aaa000000000000000000
    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:<port>/SKLM/rest/v1/keygroups/GROUP-7d588437-e725-
    48bf-a836-00a47df64e78
    Content-Type: application/json
    Accept: application/json
    Authorization: SKLMAuth userAuthId=139aeh34567m
    ```

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

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.

Procedure

1. Go 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
  a. Go to the *<WAS_HOME>*/bin directory. For example,

     Windows
     ```
     cd drive:\Program Files\IBM\WebSphere\AppServer\bin
     ```

     Linux
     ```
     cd /opt/IBM/WebSphere/AppServer/bin
     ```

  b. Start the **wsadmin** interface by using an authorized user ID, such as SKLMAdmin. For example,

     Windows
     ```
     wsadmin.bat -username SKLMAdmin -password mypwd -lang jython
     ```

     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:<port>/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"}
     ```

What to do next

Next, verify whether the drive that you added is listed in the Devices table.

Modifying a drive

You can modify information about a device such as a tape drive in the IBM Security Key Lifecycle Manager database. For example, you can 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 or REST interface, obtain the value of the uuid for the device that you intend to update.

Procedure

1. Go 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
     a. Go to the *<WAS_HOME>*/bin directory. For example,

        Windows
        ```
        cd drive:\Program Files\IBM\WebSphere\AppServer\bin
        ```

        Linux
        ```
        cd /opt/IBM/WebSphere/AppServer/bin
        ```
     b. Start the **wsadmin** interface by using an authorized user ID, such as SKLMAdmin. For example,

        Windows
        ```
        wsadmin.bat -username SKLMAdmin -password mypwd -lang jython
        ```

        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 myLTOdrive}"]')
     ```
   - 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:<port>/SKLM/rest/v1/devices?type=LTO
     Content-Type: application/json
     Accept : application/json
     Authorization: SKLMAuth userAuthId=139aeh34567m
     Accept-Language : en
     ```
     ```
     PUT https://localhost:<port>/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 myLTOdrive"}
```

What to do next

Next, 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 can 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

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.

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. Go 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
     a. Go to the *<WAS_HOME>*`/bin` directory. For example,

        Windows
        ```
        cd drive:\Program Files\IBM\WebSphere\AppServer\bin
        ```

        Linux
        ```
        cd /opt/IBM/WebSphere/AppServer/bin
        ```

     b. Start the **wsadmin** interface by using an authorized user ID, such as `SKLMAdmin`. For example,

        Windows
        ```
        wsadmin.bat -username SKLMAdmin -password mypwd -lang jython
        ```

        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:<port>/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:<port>/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
  ```

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, create certificates or certificate requests for IBM Security Key Lifecycle Manager. Before you begin, determine your site policy for the use of self-signed and certificates that are issued by a certificate authority (CA). You can create self-signed certificates for the test phase of your project. In advance, you can also request certificates from a certificate authority for the production phase.

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.

Procedure

1. Go 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
     a. Go to the <*WAS_HOME*>/bin directory. For example,

        Windows
        ```
        cd drive:\Program Files\IBM\WebSphere\AppServer\bin
        ```
        Linux
        ```
        cd /opt/IBM/WebSphere/AppServer/bin
        ```
     b. Start the **wsadmin** interface by using an authorized user ID, such as SKLMAdmin. For example,

        Windows
        ```
        wsadmin.bat -username SKLMAdmin -password mypwd -lang jython
        ```
        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 run **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:<port>/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:<port>/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"}
    ```

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 can 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 can identify a 3592 tape drive for use with IBM Security Key Lifecycle Manager. Before you begin, create the certificates that you want to associate with the devices that you are about to identify.

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.

You can make any of the following choices for serving keys to devices.

Only accept manually added devices for communication
All incoming devices are not added to the data store. You must manually specify key service to each device.

Hold new device requests pending my approval
All incoming devices of a valid device group are added to the device store, 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.

Automatically accept all new device requests for communication
All new incoming devices of a valid device group are added to the data store and are automatically served keys upon request.

Note: Do not use this setting if you intend to move the new device to another device group. Instead, select manual or pending approval mode to allow an opportunity to move the device into the appropriate device group before any keys are served.

Procedure

1. Go 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
     a. Go to the *<WAS_HOME>*/bin directory. For example,

        Windows
        ```
        cd drive:\Program Files\IBM\WebSphere\AppServer\bin
        ```

        Linux
        ```
        cd /opt/IBM/WebSphere/AppServer/bin
        ```
     b. Start the **wsadmin** interface by using an authorized user ID, such as SKLMAdmin. For example,

        Windows
        ```
        wsadmin.bat -username SKLMAdmin -password mypwd -lang jython
        ```

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 run **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:<port>/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:<port>/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"}
```

What to do next

Next, 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 can map their association, or add, modify, or delete specific certificates or devices.

About this task

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.

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.

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

1. Log on to the graphical user interface:
   a. In the Key and Device Management section on Welcome page, select **3592**.
   b. Click **Go to** > **Manage keys and devices**.
   c. 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.

2. 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 can add more certificates or certificate requests for use with IBM Security Key Lifecycle Manager. Before you begin, determine your site policy on the use of self-signed and CA certificates. You can create self-signed certificates for the test phase of your project. In advance, you can also request certificates from a certificate authority for the production phase.

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. Go 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
     a. Go to the `<WAS_HOME>/bin` directory. For example,

Windows

```
cd drive:\Program Files\IBM\WebSphere\AppServer\bin
```

Linux `cd /opt/IBM/WebSphere/AppServer/bin`

b. Start the **`wsadmin`** interface by using an authorized user ID, such as `SKLMAdmin`. For example,

Windows

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

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:<port>/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:<port>/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"}
```

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 can 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 can 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. Go 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
     a. Go to the *<WAS_HOME>*/bin directory. For example,

        Windows
        ```
        cd drive:\Program Files\IBM\WebSphere\AppServer\bin
        ```
        Linux
        ```
        cd /opt/IBM/WebSphere/AppServer/bin
        ```
     b. Start the **wsadmin** interface by using an authorized user ID, such as SKLMAdmin. For example,

        Windows
        ```
        wsadmin.bat -username SKLMAdmin -password mypwd -lang jython
        ```
        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 can modify whether a certificate is used as the system default or system partner certificate. 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.

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.

Procedure

1. Go 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
     a. Go to the *<WAS_HOME>*`/bin` directory. For example,

        Windows
        ```
        cd drive:\Program Files\IBM\WebSphere\AppServer\bin
        ```
        Linux
        ```
        cd /opt/IBM/WebSphere/AppServer/bin
        ```
     b. Start the **`wsadmin`** interface by using an authorized user ID, such as `SKLMAdmin`. For example,

        Windows
        ```
        wsadmin.bat -username SKLMAdmin -password mypwd -lang jython
        ```
        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:<port>/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:<port>/SKLM/rest/v1/certificates
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth authId=139aeh34567m
{"uuid":"CERTIFICATE-99fc36a-4ab6a0e12343","usage":
"3592","attributes":"information newinformation" }
```

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 can 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

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.

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.

Deleting a certificate deletes the material from the database.

Procedure

1. Go 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
    a. Go to the <*WAS_HOME*>/bin directory. For example,

       Windows
       ```
       cd drive:\Program Files\IBM\WebSphere\AppServer\bin
       ```

       Linux `cd /opt/IBM/WebSphere/AppServer/bin`

    b. Start the **wsadmin** interface by using an authorized user ID, such as SKLMAdmin. For example,

       Windows
       ```
       wsadmin.bat -username SKLMAdmin -password mypwd -lang jython
       ```

       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:<port>/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:<port>/SKLM/rest/v1/certificates/mycertalias
     Content-Type: application/json
     Accept : application/json
     Authorization: SKLMAuth authId=139aeh34567m
     Accept-Language : en
     ```

What to do next

Next, you might back up the keystore again to accurately reflect the change in certificates.

Adding a drive

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

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.

Procedure

1. Go 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
     a. Go to the *<WAS_HOME>*/bin directory. For example,

        Windows
        ```
        cd drive:\Program Files\IBM\WebSphere\AppServer\bin
        ```

        Linux
        ```
        cd /opt/IBM/WebSphere/AppServer/bin
        ```

     b. Start the **wsadmin** interface by using an authorized user ID, such as SKLMAdmin. For example,

        Windows
        ```
        wsadmin.bat -username SKLMAdmin -password mypwd -lang jython
        ```

        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:<port>/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"}
     ```

What to do next

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

Modifying a drive

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

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.

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.

Procedure

1. Go 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
     a. Go to the `<`*`WAS_HOME`*`>/bin` directory. For example,

        Windows
        ```
        cd drive:\Program Files\IBM\WebSphere\AppServer\bin
        ```

        Linux
        ```
        cd /opt/IBM/WebSphere/AppServer/bin
        ```
     b. Start the **`wsadmin`** interface by using an authorized user ID, such as `SKLMAdmin`. For example,

        Windows
        ```
        wsadmin.bat -username SKLMAdmin -password mypwd -lang jython
        ```

        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:<port>/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:<port>/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"}
  ```

What to do next

Next, 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 can 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

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.

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. Go 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

a. Go to the *<WAS_HOME>*/bin directory. For example,

Windows

```
cd drive:\Program Files\IBM\WebSphere\AppServer\bin
```

Linux

```
cd /opt/IBM/WebSphere/AppServer/bin
```

b. Start the **wsadmin** interface by using an authorized user ID, such as SKLMAdmin. For example,

Windows

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

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:<port>/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:<port>/SKLM/rest/v1/devices/DEVICE-74386920-148c-
47b2-a1e2-d19194b315cf
Content-Type: application/json
Authorization: SKLMAuth authId=139aeh34567m
Accept : application/json
```

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, 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. Go 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
     a. Go to the `<`*`WAS_HOME`*`>/bin` directory. For example,

        Windows
        ```
        cd drive:\Program Files\IBM\WebSphere\AppServer\bin
        ```
        Linux
        ```
        cd /opt/IBM/WebSphere/AppServer/bin
        ```
     b. Start the **`wsadmin`** interface by using an authorized user ID, such as `SKLMAdmin`. For example,

        Windows
        ```
        wsadmin.bat -username SKLMAdmin -password mypwd -lang jython
        ```
        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:<port>/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:<port>/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"}
    ```

What to do next

Next, 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

Identify a storage image (device) for use with IBM Security Key Lifecycle Manager. Before you begin, create the image certificates that you want to associate with the storage images that you are about to identify.

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.

You can make any of the following choices for serving keys to devices.

Only accept manually added devices for communication
All incoming devices are not added to the data store. You must manually specify key service to each device.

Hold new device requests pending my approval
All incoming devices of a valid device group are added to the device store, 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.

Automatically accept all new device requests for communication
All new incoming devices of a valid device group are added to the data store and are automatically served keys upon request.

Note: Do not use this setting if you intend to move the new device to another device group. Instead, select manual or pending approval mode to allow an opportunity to move the device into the appropriate device group before any keys are served.

Procedure

1. Go 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

     a. Go to the *<WAS_HOME>*/bin directory. For example,

     Windows
     ```
     cd drive:\Program Files\IBM\WebSphere\AppServer\bin
     ```
     Linux
     ```
     cd /opt/IBM/WebSphere/AppServer/bin
     ```
     b. Start the **wsadmin** interface by using an authorized user ID, such as SKLMAdmin. For example,

     Windows
     ```
     wsadmin.bat -username SKLMAdmin -password mypwd -lang jython
     ```

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 run **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:<port>/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:<port>/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"}
```

What to do next

Next, you can 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 can map their association, or add, modify, or delete specific certificates or storage images.

About this task

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.

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.

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.
❌	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

1. Log on to the graphical user interface.
   a. In the Key and Device Management section on Welcome page, select **DS8000**.
   b. Click **Go to** > **Manage keys and devices**.

c. 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.

2. On the DS8000 Key and Device Management page, you can add, modify, or delete a storage image or image certificate.

   You can do the following 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 can add more image certificates or certificate requests for use with IBM Security Key Lifecycle Manager. Before you begin, determine your site policy on the use of certificates.

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. Go 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
     a. Go to the `<*WAS_HOME*>/bin` directory. For example,

Windows
```
cd drive:\Program Files\IBM\WebSphere\AppServer\bin
```

Linux
```
cd /opt/IBM/WebSphere/AppServer/bin
```

b. Start the **wsadmin** interface by using an authorized user ID, such as SKLMAdmin. For example,

Windows

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

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:<port>/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:<port>/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"}
```

What to do next

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

- Certificate:

  You can 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 can 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. Go 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
     a. Go to the *`<WAS_HOME>`*`/bin` directory. For example,

        Windows

        ```
        cd drive:\Program Files\IBM\WebSphere\AppServer\bin
        ```

Linux `cd /opt/IBM/WebSphere/AppServer/bin`

b. Start the **`wsadmin`** interface by using an authorized user ID, such as `SKLMAdmin`. For example,

Windows

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

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:<port>/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:<port>/SKLM/rest/v1/certificates
     Content-Type: application/json
     Accept: application/json
     Authorization: SKLMAuth authId=139aeh34567m
     {"uuid":"CERTIFICATE-33fc26e-5fb5a0e66143","usage":
     "DS8000","attributes":"information {newinformation}" }
     ```

What to do next

Next, you can use the DS8000 Key and Device Management page to associate image certificates with specific storage images.

Deleting an image certificate

You can 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

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.

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.

Deleting a certificate deletes the material from the database.

Procedure

1. Go 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
     a. Go to the *<WAS_HOME>*/bin directory. For example,

        Windows

        ```
        cd drive:\Program Files\IBM\WebSphere\AppServer\bin
        ```

        Linux

        ```
        cd /opt/IBM/WebSphere/AppServer/bin
        ```

     b. Start the **wsadmin** interface by using an authorized user ID, such as SKLMAdmin. For example,

        Windows

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

        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:<port>/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:<port>/SKLM/rest/v1/certificates/mycertalias
  Content-Type: application/json
  Accept: application/json
  Authorization: SKLMAuth authId=139aeh34567m
  Accept-Language: en
  ```

What to do next

Next, you can back up the keystore again to accurately reflect the change in certificates.

Adding a storage image

You can add a storage image to the IBM Security Key Lifecycle Manager database. 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.

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.

Procedure

1. Go 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
     a. Go to the *<WAS_HOME>*`/bin` directory. For example,

        Windows
        ```
        cd drive:\Program Files\IBM\WebSphere\AppServer\bin
        ```

        Linux
        ```
        cd /opt/IBM/WebSphere/AppServer/bin
        ```

     b. Start the **wsadmin** interface by using an authorized user ID, such as `SKLMAdmin`. For example,

        Windows
        ```
        wsadmin.bat -username SKLMAdmin -password mypwd -lang jython
        ```

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:<port>/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"}
     ```

What to do next

Next, you can determine the status of the storage image that you added.

Modifying a storage image

You can 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

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.

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.

Procedure

1. Go 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
   a. Go to the *<WAS_HOME>*/bin directory. For example,

      Windows
      ```
      cd drive:\Program Files\IBM\WebSphere\AppServer\bin
      ```

      Linux
      ```
      cd /opt/IBM/WebSphere/AppServer/bin
      ```

   b. Start the **wsadmin** interface by using an authorized user ID, such as SKLMAdmin. For example,

      Windows
      ```
      wsadmin.bat -username SKLMAdmin -password mypwd -lang jython
      ```

      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:<port>/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"}
     ```

Deleting a storage image

You can 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

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.

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.

Procedure

1. Go 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

a. Go to the `<WAS_HOME>/bin` directory. For example,

Windows
```
cd drive:\Program Files\IBM\WebSphere\AppServer\bin
```

Linux `cd /opt/IBM/WebSphere/AppServer/bin`

b. Start the **`wsadmin`** interface by using an authorized user ID, such as `SKLMAdmin`. For example,

Windows
```
wsadmin.bat -username SKLMAdmin -password mypwd -lang jython
```

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:<port>/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:<port>/SKLM/rest/v1/devices/DEVICE-74386920-148c-
47b2-a1e2-d19194b315cf
Content-Type: application/json
Accept : application/json
Authorization: SKLMAuth authId=139aeh34567m
```

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 the following 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 can do the following 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.
     - More Keys

       Select a device and then select **Add > More Keys**. On the Add Key dialog, specify the necessary 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 you select the **Machine affinity** check box on the Key and Device Management page, value of the **`device.enableMachineAffinity`** property is set to true. Using machine affinity, you can set key serving for specific device and machine combinations.

When machine affinity is enabled, use the Add Association dialog to specify the necessary 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 can 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. Go 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

a. Go to the `<WAS_HOME>/bin` directory. For example,

Windows

```
cd drive:\Program Files\IBM\WebSphere\AppServer\bin
```

Linux

```
cd /opt/IBM/WebSphere/AppServer/bin
```

b. Start the **`wsadmin`** interface by using an authorized user ID, such as `SKLMAdmin`. For example,

Windows

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

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 30423830303030343700000000000}"]')
```

- 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:<port>/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"}
```

What to do next

Next, you can associate the device with a machine.

Modifying a device

You can 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. Go 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
     a. Go to the `<`*`WAS_HOME`*`>/bin` directory. For example,

        Windows
        ```
        cd drive:\Program Files\IBM\WebSphere\AppServer\bin
        ```
        Linux
        ```
        cd /opt/IBM/WebSphere/AppServer/bin
        ```
     b. Start the **wsadmin** interface by using an authorized user ID, such as `SKLMAdmin`. For example,

        Windows
        ```
        wsadmin.bat -username SKLMAdmin -password mypwd -lang jython
        ```
        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:<port>/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"}
```

What to do next

Next, you might verify that the changes are made.

Deleting a device

You can 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. Go 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
     a. Go to the *<WAS_HOME>*/bin directory. For example,

        Windows
        ```
        cd drive:\Program Files\IBM\WebSphere\AppServer\bin
        ```

        Linux
        ```
        cd /opt/IBM/WebSphere/AppServer/bin
        ```
     b. Start the **wsadmin** interface by using an authorized user ID, such as SKLMAdmin. For example,

        Windows
        ```
        wsadmin.bat -username SKLMAdmin -password mypwd -lang jython
        ```

        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 3042383030303437000000000000]')
```

```
print AdminTask.tklmMachineDeviceDelete
('[-deviceUUID DEVICE-663b6d37-e6d5-4c9f-af90-e40e48d27f3c
 -machineID 3042383030303437000000000000]')
```

You can send the following HTTP requests:

```
GET https://localhost:<port>/SKLM/rest/v1/machines/device?machineID=
3042383030303437000000000000
Content-Type: application/json
Accept : application/json
Authorization: SKLMAuth authId=139aeh34567m
```

```
DELETE https://localhost:<port>/SKLM/rest/v1/machines/device
Content-Type: application/json
Accept : application/json
Authorization: SKLMAuth authId=139aeh34567m
{"deviceUUID":"DEVICE-663b6d37-e6d5-4c9f-af90-e40e48d27f3c","machineID":
"3042383030303437000000000000"}
```

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:<port>/SKLM/rest/v1/devices/DEVICE-74386920-148c-
     47b2-a1e2-d19194b315cf
     Content-Type: application/json
     Accept : application/json
     Authorization: SKLMAuth authId=139aeh34567m
     ```

Adding keys

You can add more keys for use with DS5000 storage servers. Before you begin, determine your site policy for naming key prefixes.

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.

Procedure

1. Go 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
     a. Go to the `<`*`WAS_HOME`*`>/bin` directory. For example,

        Windows
        ```
        cd drive:\Program Files\IBM\WebSphere\AppServer\bin
        ```

        Linux
        ```
        cd /opt/IBM/WebSphere/AppServer/bin
        ```
     b. Start the **`wsadmin`** interface by using an authorized user ID, such as `SKLMAdmin`. For example,

        Windows
        ```
        wsadmin.bat -username SKLMAdmin -password mypwd -lang jython
        ```

        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: dsk000000000000000001
          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:
  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 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:<port>/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)": "dsk000000000000000001",
  "key store name(s)": "defaultKeyStore "
.
.
.
```

  c. 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:<port>/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 can associate the device with a machine.

Modifying (deleting) keys

You can modify the number of keys that a DS5000 storage server uses by deleting one or more keys. Before you begin, determine the changed information, such as the number of keys that you want to delete.

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.

Procedure

1. Go 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
     a. Go to the <*WAS_HOME*>/bin directory. For example,

        Windows
        ```
        cd drive:\Program Files\IBM\WebSphere\AppServer\bin
        ```
        Linux
        ```
        cd /opt/IBM/WebSphere/AppServer/bin
        ```
     b. Start the **wsadmin** interface by using an authorized user ID, such as SKLMAdmin. For example,

        Windows
        ```
        wsadmin.bat -username SKLMAdmin -password mypwd -lang jython
        ```
        Linux
        ```
        ./wsadmin.sh -username SKLMAdmin -password mypwd -lang jython
        ```
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 aaa000000000000000000
      -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:<port>/SKLM/rest/v1/keys/aaa000000000000000000
  Content-Type: application/json
  Accept : application/json
  Authorization: SKLMAuth authId=139aeh34567m
  ```

What to do next

Next, you might associate the device with a machine.

GPFS (IBM Spectrum Scale) file management

You can use GPFS file system to manage keys in IBM Security Key Lifecycle Manager.

GPFS (IBM Spectrum Scale) 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.

GPFS provides support for file encryption that ensures both secure storage and secure deletion of data. GPFS manages encryption through the use of encryption keys and encryption policies.

See GPFS documentation for more information http://www.ibm.com/support/knowledgecenter/SSFKCN/gpfs_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 GPFS 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 **GPFS**.
   b. Click **Go to** > **Manage keys and devices**.

   c. Alternatively, right-click **GPFS** 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 modify the device group that a certificate or key belongs to, select the certificate or key and then click **Modify**. Alternatively, right-click the selected certificate or key. Then, click **Modify**.

     – 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.

     – Key

       View read-only information in the Modify Key page. 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 able.

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 **GPFS**.

3. Click **Go to** > **Manage keys and devices**.
4. Alternatively, right-click **GPFS** and select **Manage keys and devices**.
5. On the management page for `GPFS`, 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 **GPFS**.
3. Click **Go to** > **Manage keys and devices**.
4. Alternatively, right-click **GPFS** and select **Manage keys and devices**.
5. On the management page for `GPFS`, 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 **GPFS**.
3. Click **Go to** > **Manage keys and devices**.
4. Alternatively, right-click **GPFS** and select **Manage keys and devices**.
5. On the management page for `GPFS`, 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 GPFS.

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 **GPFS**.
3. Click **Go to** > **Manage keys and devices**.
4. Alternatively, right-click **GPFS** and select **Manage keys and devices**.
5. On the management page for `GPFS`, 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 **GPFS**.
3. Click **Go to** > **Manage keys and devices**.
4. Alternatively, right-click **GPFS** and select **Manage keys and devices**.
5. On the management page for `GPFS`, 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 **GPFS**.
3. Click **Go to** > **Manage keys and devices**.
4. Alternatively, right-click **GPFS** and select **Manage keys and devices**.
5. On the management page for `GPFS`, 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.

PEER_TO_PEER management

The PEER_TO_PEER device group serves the devices that operate on Key Management Interoperability Protocol (KMIP). This device group allows a maximum of two devices to share one or more symmetric keys.

You can use the PEER_TO_PEER device group to manage keys and device certificates in IBM Security Key Lifecycle Manager.

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 PEER_TO_PEER 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 the device WWNN, device certificate name, and device type are displayed.
- In right columns, information about keys indicates the key UUID and the key name that the certificates on the left have access to, are displayed.

Procedure

1. Log on to the graphical user interface.
   a. In the Key and Device Management section on Welcome page, select **PEER_TO_PEER**.
   b. Click **Go to** > **Manage keys and devices**.

   c. Alternatively, right-click **PEER_TO_PEER** and select **Manage keys and devices**.
2. You can add, modify, or delete a key or device certificate.

Adding a device

You can add a device to the PEER_TO_PEER device group for use with IBM Security Key Lifecycle Manager.

About this task

You can use the Add Device dialog box to add a device. 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 **PEER_TO_PEER**.
3. Click **Go to** > **Manage keys and devices**.
4. Alternatively, right-click **PEER_TO_PEER** and select **Manage keys and devices**.
5. On the management page for `PEER_TO_PEER`, click **Add**.
6. Click **Device**.
7. On the Add Device dialog, specify the device certificate name and location, and the device type. You can add only one device of each type, and a maximum of two devices.
8. Click **Add**.

   The device is added to the PEER_TO_PEER table.

Modifying a device

You can modify a device certificate in the IBM Security Key Lifecycle Manager database.

About this task

Use the Modify Device Certificate dialog box to update a device. 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 **PEER_TO_PEER**.
3. Click **Go to** > **Manage keys and devices**.
4. Alternatively, right-click **PEER_TO_PEER** and select **Manage keys and devices**.
5. On the management page for `PEER_TO_PEER`, select a device.
6. Click **Modify**.
7. Alternatively, right-click a device and then select **Modify**, or double-click a device entry.
8. On the Modify Device Certificate dialog box, select a certificate that has the same WWNN as the earlier device certificate.
9. Click **Modify**.

   The device information is changed in the table.

What to do next

Next, you might verify that the changes are made.

Deleting a device

You might delete a selected device and its corresponding communication certificate, which can be in any state, such as active. After deletion, this device cannot communicate with the objects of the PEER_TO_PEER group.

About this task

You can use the `Delete` menu item to delete a device. 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 **PEER_TO_PEER**.
3. Click **Go to** > **Manage keys and devices**.
4. Alternatively, right-click **PEER_TO_PEER** and select **Manage keys and devices**.
5. On the management page for `PEER_TO_PEER`, select a device.
6. Click **Delete**.
7. Alternatively, right-click a device and then select **Delete**.
8. On the Confirm dialog, read the confirmation message. Then, click **OK**.

   The device is removed from the table.

Adding keys

You might add keys for use with PEER_TO_PEER.

About this task

You can use the Add Key dialog box 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 **PEER_TO_PEER**.
3. Click **Go to** > **Manage keys and devices**.
4. Alternatively, right-click **PEER_TO_PEER** and select **Manage keys and devices**.
5. On the management page for `PEER_TO_PEER`, 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 box 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 **PEER_TO_PEER**.
3. Click **Go to** > **Manage keys and devices**.
4. Alternatively, right-click **PEER_TO_PEER** and select **Manage keys and devices**.
5. On the management page for `PEER_TO_PEER`, 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 box, 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 **PEER_TO_PEER**.
3. Click **Go to** > **Manage keys and devices**.
4. Alternatively, right-click **PEER_TO_PEER** and select **Manage keys and devices**.
5. On the management page for `PEER_TO_PEER`, 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.

Export and import of device groups

IBM Security Key Lifecycle Manager provides a set of operations to export the device groups from one instance of IBM Security Key Lifecycle Manager and import it into another instance that has the same version as of the source IBM Security Key Lifecycle Manager instance across operating systems. The exported device group data is encrypted and protected through a password.

For more information about device group import and export, see Overview of device group export and import

Exporting a device group

You can export device group data for the selected device group to an encrypted archive. Then, you can import this device group data into another instance of IBM Security Key Lifecycle Manager across operating systems.

About this task

You can use the Export Device Group dialog to export a device group. Alternatively, you can use **`Device Group Export REST Service`**.

Your role must have a permission to export device groups.

Note: During data migration from previous versions of IBM Security Key Lifecycle Manager, some of the certificates might not be associated with the correct device group. As a result, it is possible that a few certificates are falsely shown (in UI, REST, or CLI) for a device group, such as 3592 or DS8000, even though the certificates do not belong to the device group. When you export such device groups, only the certificates of the device group are exported. The falsely shown certificates are not exported.

Procedure

1. Go 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 a device group.

   c. Click **Go to** > **Export**.

   d. Alternatively, right-click the selected device group and select **Export**.

   e. Alternatively, on the Welcome page, click **Administration** > **Export and Import** > **Export**.

   REST interface

   Open a REST client.

2. Export the device group data for the selected device group to the directory you specified.

   Graphical user interface

   a. On the Export Device Group dialog, the **Device Group** field specifies the selected device group.

   b. To change the device group, click **Select**.

   c. The **Export repository location** field displays the default <*SKLM_DATA*> directory path, where the export file is saved, for example, `C:\Program Files\IBM\WebSphere\AppServer\products\sklm\data`. For the definition of <*SKLM_DATA*>, see Definitions for *HOME* and other directory variables. Click **Browse** to specify a export repository location under <*SKLM_DATA*> directory.

      Directory path in the **Export repository location** field changes based on the value that is set for the **`browse.root.dir`** property in the `SKLMConfig.properties` file.

   d. In the **Password** field, specify a value for the encryption password. Ensure that you retain the encryption password for future use.

   e. In the **Retype password** field, retype the password that you entered in the **Password** field.

f. In the **Description** field, specify additional information that indicates the purpose of the device group export file.

g. Click **Export**.

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 run **`Device Group Export 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:<port>/SKLM/rest/v1/ckms/deviceGroupsExport
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
{"name": "3592", "exportDirectory": "/opt/IBM/WebSphere/AppServer/products/sklm/data/
"password": "mypassword"}
```

3. When the export process is complete, a message box is displayed to indicate that the export operation is complete.

What to do next

Ensure that you retain this password for use when you later import and decrypt the device group export file into another instance of IBM Security Key Lifecycle Manager. Review the directory that contains the export archive to ensure that the export file exists. You can also verify whether the archive is listed in the table on the **IBM Security Key Lifecycle Manager** > **Administration** > **Export and Import** > **Export/Import** page.

Importing a device group

You can import device group data that were exported from another instance of IBM Security Key Lifecycle Manager if you want to move data across IBM Security Key Lifecycle Manager instances.

Before you begin

You must have the export file and ensure that you have the password that you used when the export file was created. Save the export files in the default <*SKLM_DATA*> directory, for example, `C:\Program Files\IBM\WebSphere\AppServer\products\sklm\data`. For the definition of <*SKLM_DATA*>, see Definitions for *HOME* and other directory variables.

The <*SKLM_DATA*> directory path changes based on the value that is set for the **`browse.root.dir`** property in the `SKLMConfig.properties` file.

Version of the IBM Security Key Lifecycle Manager instance where the device group export data is being imported must be same as the IBM Security Key Lifecycle Manager instance from which the device group data were exported.

About this task

At times, the device group data that is imported might conflict with an existing data in the database. For example, a key in the imported device group might be a key with same alias name of a device group in the current instance of IBM Security

Key Lifecycle Manager where the data is being imported. When conflicts occur, they must be resolved before the import process can continue.

You can use the Export and Import page. Alternatively, you can use `Device Group Import REST Service` to import device groups.

Your role must have a permission to import device groups. For more information about device group export and import operations, see Overview of device group export and import.

Procedure

1. Go to the appropriate page or directory.

   Graphical user interface

   a. Log on to the graphical user interface.
   b. On the Welcome page, click **Administration** > **Export and Import**.

   REST interface

   Open a REST client.

2. Import a selected export file. Only one export or import task can run at a time. If you want import a file to an IBM Security Key Lifecycle Manager instance on a different system, copy the export file to that system by using media such as a disk, or electronic transmission.

   Graphical user interface

   a. Click **Browse** to specify the export file location under <SKLM_DATA> directory, for example, `C:\Program Files\IBM\WebSphere\AppServer\products\sklm\data`.
   b. Click **Display Exports** to display the export files.
   c. In the table, select an export file.
   d. Click **Import**.
   e. Alternatively, double-click or right-click the export file and select **Import**.
   f. On the Import from Export Archive dialog, specify the encryption password that you used to create the export file.
   g. Click **Import** to start the import operation.

   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 run **`Device Group Import 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:<port>/SKLM/rest/v1/ckms/deviceGroupsImport
      Content-Type: application/json
      Accept: application/json
      Authorization: SKLMAuth userAuthId=139aeh34567m
      {"importFilePath": "C:\\Program Files\\IBM\\WebSphere\\AppServer\\products\\sklm\\c
      "password": "passw0rd123"}
      ```

3. If any conflicts arise during the import process, the Conflicts while Importing dialog appears. See the “Viewing the import conflicts” on page 111 topic for more information.

4. If no data conflicts, the progress dialog box appears. When the import process is complete, a message box is displayed to indicate that the import operation is complete.
5. Click **Close**.
6. Restart the server. For instructions about how to stop and start the server, see "Restarting the IBM Security Key Lifecycle Manager server" on page 190.

Deleting a device group export file

You might delete an export file for which a business needs no longer exists. Use the graphical user interface or REST interface to delete a device group export file.

About this task

You can use the Export/Import page or **`Device Group Delete REST Service`** to delete an export file.

Procedure

1. Go to the appropriate page or directory.

   Graphical user interface

   a. Log on to the graphical user interface.

   b. On the Welcome page, click **Administration** > **Export and Import**.

   REST interface

   Open a REST client.

2. Delete a selected export file.

   Graphical user interface

   a. Click **Browse** to specify the export file location under <SKLM_DATA> directory. For the definition of `<SKLM_DATA>`, see Definitions for *HOME* and other directory variables.

   b. Click **Display Exports** to display the export files.

   c. In the table, select an export file.

   d. Click **Delete** and confirm that you want to delete the file.

   e. Alternatively, right-click the export file and select **Delete** and confirm that you want to delete the file.

   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 run **`Device Group Delete REST Service`**, send the HTTP DELETE request. Pass the user authentication identifier that you obtained in `Step a` along with the request message as shown in the following example.

```
DELETE https://localhost:<port>/SKLM/rest/v1/deviceGroups/newDevGrp
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
```

What to do next

Examine the directory in which the export files are stored to determine whether the specified file was deleted.

Viewing the import conflicts

When the device group data is imported from an export file, its content is analyzed for conflicts with the data that is stored in the IBM Security Key Lifecycle Manager database. The conflicts must be resolved before the data can be imported. You can view the list of conflicts to analyze and resolve the problems.

About this task

Import conflict details are opened in the Conflicts while Importing window. You can export the conflicts data in comma-separated value (CSV) format.

Procedure

1. Go to the appropriate page or directory.

   Graphical user interface

   a. Log on to the graphical user interface.

   b. On the Welcome page, click **Administration** > **Export and Import**.

   REST interface

   Open a REST client.

2. Import a selected export file. Only one export or import task can run at a time. If you want import a file to an IBM Security Key Lifecycle Manager instance on a different system, copy the export file to that system by using media such as a disk, or electronic transmission.

   Graphical user interface

   a. Click **Browse** to specify the export file location under <SKLM_DATA> directory, for example, `C:\Program Files\IBM\WebSphere\AppServer\products\sklm\data`. For the definition of `<SKLM_DATA>`, see Definitions for *HOME* and other directory variables.

   b. Click **Display Exports** to display the export files.

   c. In the table, select an export file.

   d. Click **Import**.

   e. Alternatively, double-click or right-click the export file and select **Import**.

   f. On the Import from Export Archive dialog, specify the encryption password that you used to create the export file.

   g. Click **Import** to start the import operation.

   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 run **`Device Group Import 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:<port>/SKLM/rest/v1/ckms/deviceGroupsImport
   Content-Type: application/json
   Accept: application/json
   Authorization: SKLMAuth userAuthId=139aeh34567m
   {"importFilePath": "C:\\Program Files\\IBM\\WebSphere\\AppServer\\products\\sklm\\c
   "password": "passw0rd123"}
   ```

3. If the conflicts are detected during import operation, you can view the list of conflicts in the Conflicts while Importing window.
4. You can also use **Device Group Import Conflicts REST Service** to list the data conflicts, if any, when the device group data is imported from an export file into an IBM Security Key Lifecycle Manager instance. To run **Device Group Import Conflicts REST Service**, send the HTTP POST request.

   ```
   POST https://localhost:<port>/SKLM/rest/v1/ckms/deviceGroupsImport/importConflicts
   Content-Type: application/json
   Accept: application/json
   Authorization: SKLMAuth userAuthId=139aeh34567m
   {"importFilePath": "C:\\Program Files\\IBM\\WebSphere\\AppServer\\products\\sklm\\data\\sklm_v2.7
   "password": "passw0rd123"}
   ```
5. To export the import conflicts data to a file in comma-separated values (CSV) format for further analysis, click **Export Conflict Report**.

What to do next

You must resolve the conflicts before the data can be imported. You can use the following REST services to resolve import conflicts:

- **Change Name REST Service**
- **Change Certificate Alias REST Service**
- **Change History REST Service**
- **Renew Key Alias REST Service**

Viewing device group export and import history

Use the History page to view details of all the device group export and import operations that are run on the current instance of IBM Security Key Lifecycle Manager.

Procedure

1. Log on to the graphical user interface.
2. On the Welcome page, click **Administration** > **Export and Import**.
3. In the Export and Import page, click **History**.

   A list of device group export and import operations that run on the current instance of IBM Security Key Lifecycle Manager is displayed.
4. Select a row and double-click. Summary details of the export or import operation is displayed.

Viewing device group export and import summary information

Use the Export/Import Summary page to view the details of a selected export file for understanding and working with the device group data. You can view the details of an export file that is created in the current instance of IBM Security Key Lifecycle Manager. You can also view details of an export file that you want to import into the current instance of IBM Security Key Lifecycle Manager.

Procedure

1. Log on to the graphical user interface.
2. On the Welcome page, click **Administration** > **Export and Import**.
3. Select a device group export file that is listed in the table.
4. Click **Summary**.
5. Alternatively, right-click the export file and select **Summary**.

The following table provides the summary information.

ID	ID of the selected device group export file.
Archive Name	Name of the device group export file.
Start Time	Time at which the export or import operation was started.
End Time	Time at which the export or import operation was ended.
Device Group	Name of the device group from which the data is exported to the file.
<device group data>	Displays the number of certificates, key groups, and other details in the device group export file.

6. Click **Cancel** to close the summary page.

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 and IBM Tivoli 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: In IBM Security Key Lifecycle Manager, Version 3.0 and later, 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 3.0 or later 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.

Encryption methods to back up IBM Security Key Lifecycle Manager data

IBM Security Key Lifecycle Manager supports the following encryption methods for backups:

Password-based encryption
During the backup process, a password is specified to encrypt the backup key, and you must specify the same encryption password to decrypt and restore the backup files.

HSM-based encryption
You can configure IBM Security Key Lifecycle Manager to use Hardware Security Module (HSM) for storing the master encryption key. During the backup process, the backup key is encrypted by the master key, which is stored in HSM. During the restore process, the master key in HSM decrypts the backup key. Then, the backup key is used to restore backup contents.

High-performance backup and restore

High-performance backup and restore provide backup and restoration of large amounts of encryption keys. You can configure IBM Security Key Lifecycle Manager for high-performance backup and restore operations by setting the following parameter in the `SKLMConfig.properties` configuration file.

```
enableHighScaleBackup=true
```

When IBM Security Key Lifecycle Manager is configured for high-performance backup and restore, IBM DB2 native backup technology is used to run the backup and restore operation for more efficiency. However, with this configuration, you can restore the backup only in an identical operating environment. The operating system, middleware components, and directory structures must be identical on both systems.

You cannot create a cross-platform compatible backup file if IBM Security Key Lifecycle Manager is configured for high-performance backup and restore activities. For information about how to back up large amount of data, see "Backing up large amount of data" on page 121.

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 for the password-based encryption backups, ensure that you have the password that was used when the backup file was created.
- Use the following guidelines to restore HSM-based encryption backups:
  - Ensure that the same HSM partition is present with all its key entries intact on the system where the backup file is restored.
  - Master key that was used for the backup key encryption must be intact to restore the backup file. If the master key is refreshed, all the older backups are inaccessible or unusable.
  - You must connect to the same HSM and the master key for backup and restore operations irrespective of whether you use HSM-based encryption or password-based encryption.
- 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 data with password-based encryption

You must specify an encryption password to back up IBM Security Key Lifecycle Manager data. Use the same password to decrypt and restore the backup files.

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. Go to the appropriate page or directory.

   Graphical user interface

   a. Log on to the graphical user interface.

   b. On the Welcome page, click **Administration** > **Backup and Restore**.

   Command-line interface

   a. Go to the WAS_HOME/bin directory. For example,

      Windows
      ```
      cd drive:\Program Files\IBM\WebSphere\AppServer\bin
      ```
      Linux
      ```
      cd /opt/IBM/WebSphere/AppServer/bin
      ```

   b. Start the **wsadmin** interface by using an authorized user ID, such as SKLMAdmin. For example,

Windows

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

Linux

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

REST interface

Open a REST client.

2. Create a backup file. You can run only one backup or restore task at a time.

   Graphical user interface

   a. On the **Backup and Restore** table, the **Backup repository location** field displays the default <*SKLM_DATA*> directory path, where the backup file is saved, for example, `C:\Program Files\IBM\WebSphere\AppServer\products\sklm\data`. For the definition of <*SKLM_DATA*>, see Definitions for *HOME* and other directory variables. Click **Browse** to specify a backup repository location under <*SKLM_DATA*> directory.

      Directory path in the **Backup repository location** field changes based on the value that you set for the **`tklm.backup.dir`** property in the `SKLMConfig.properties` file.

   b. Click **Create Backup**.

   c. On the Create Backup page, specify information such as a value for the encryption password and backup description. A read-only backup file location is displayed in the **Backup location** field. 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`**, the backup location, encryption password, and any other necessary information to create a backup file. For example:

   ```
   print AdminTask.tklmBackupRun ('[-backupDirectory C:\\sklmbackup1 -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 run **`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:<port>/SKLM/rest/v1/ckms/backups
      Content-Type: application/json
      Accept : application/json
      Authorization: SKLMAuth authId=139aeh34567m
      Accept-Language : en
      {"backupDirectory":"/sklmbackup1","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_v3.0.1.0_20170123144220`**`-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.

Backing up data with password-based encryption when HSM is configured

You must set the **enablePBEInHSM=true** property in the SKLMConfig.properties file to back up data with password-based encryption when Hardware Security Module (HSM) is configured.

Before you begin

Ensure that IBM Security Key Lifecycle Manager is configured to use HSM for storing the master key by using steps in the “Configuring HSM parameters” on page 194 topic.

About this task

When HSM is configured, during the backup process, the master key in HSM encrypts the backup key. HSM-based encryption is the default method for the backups when HSM is configured to store the master key. For information about HSM-based encryption, see HSM-based encryption for backups. Your role must have a permission to back up files.

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. Set the **enablePBEInHSM=true** property in the <*SKLM_HOME*>/config/SKLMConfig.properties file.

   Command-line interface

   a. Go to the WAS_HOME/bin directory. For example,

      Windows

      ```
      cd drive:\Program Files\IBM\WebSphere\AppServer\bin
      ```

      Linux

      ```
      cd /opt/IBM/WebSphere/AppServer/bin
      ```

   b. Start the **wsadmin** interface by using an authorized user ID, such as SKLMAdmin. For example,

      Windows

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

      Linux

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

   c. Run the **tklmConfigUpdateEntry** CLI command to set **enablePBEInHSM** property in the SKLMConfig.properties configuration file.

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

REST interface

a. Open a REST client.

b. 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.

c. Run **Update Config Property REST Service** to set **enablePBEInHSM** property in the SKLMConfig.properties configuration file. Pass the user authentication identifier that you obtained in Step b along with the request message as shown in the following example.

```
PUT https://localhost:<port>/SKLM/rest/v1/configProperties
Content-Type: application/json
Accept : application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
Accept-Language : en
{ "enablePBEInHSM" : "true"}
```

2. Go to the appropriate page or directory for backing up data.

Graphical user interface

a. Log on to the graphical user interface.

b. On the Welcome page, click **Administration** > **Backup and Restore**.

Command-line interface

a. Go to the WAS_HOME/bin directory.

b. Start the **wsadmin** interface by using an authorized user ID, such as SKLMAdmin.

REST interface

Open a REST client.

3. Create a backup file. You can run only one backup or restore task at a time.

Graphical user interface

a. On the **Backup and Restore** table, the **Backup repository location** field displays the default *<SKLM_DATA>* directory path, where the backup file is saved, for example, C:\Program Files\IBM\WebSphere\AppServer\products\sklm\data. For the definition of *<SKLM_DATA>*, see Definitions for *HOME* and other directory variables. Click **Browse** to specify a backup repository location under *<SKLM_DATA>* directory.

Directory path in the **Backup repository location** field changes based on the value that you set for the **tklm.backup.dir** property in the SKLMConfig.properties file.

b. Click **Create Backup**.

c. On the Create Backup page, specify information such as a value for the encryption password and backup description. A read-only backup file location is displayed in the **Backup location** field. 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**, the backup location, password, and any other necessary information to create a backup file as shown in the following example.

```
print AdminTask.tklmBackupRun
 ('[-backupDirectory C:\\sklmbackup1 -password myBackupPwd]')
```

REST interface

Run **Backup Run REST Service** by sending the HTTP POST request as shown in the following example.

```
POST https://localhost:<port>/SKLM/rest/v1/ckms/backups
Content-Type: application/json
Accept : application/json
Authorization: SKLMAuth authId=139aeh34567m
Accept-Language : en
{"backupDirectory":"/sklmbackup1","password":"myBackupPwd"}
```

4. 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_v3.0.1.0_20170123144220`**`-0800`**`_backup.jar`, where -0800 indicates that the timezone is eight hours behind GMT.

What to do next

Do not edit a file in the backup JAR file. The file that you attempt to edit becomes unreadable. You must connect to the same HSM and the master key for backup and restore operations irrespective of whether you use HSM-based encryption or password-based encryption.

Backing up data with HSM-based encryption

When IBM Security Key Lifecycle Manager is configured with Hardware Security Module (HSM) for storing the master encryption key, you can use HSM-based encryption for creating secure backups.

Before you begin

Ensure that IBM Security Key Lifecycle Manager is configured to use HSM for storing the master key before you back up data with HSM-based encryption. For the configuration steps, see “Configuring HSM parameters” on page 194.

You must consider the following guidelines for HSM-based encryption

- The same HSM partition must be present with all its key entries on the system where the backup file is restored.
- Master key that you used for the backup key encryption must be intact to restore the backup file. If the master key is refreshed, all the older backups are inaccessible or unusable.
- You must connect to the same HSM and the master key for backup and restore operations irrespective of whether you use HSM-based encryption or password-based encryption.

About this task

When you run the IBM Security Key Lifecycle Manager backup operation, a backup archive is created. The backup key in the archive encrypts backup contents. The master key in HSM encrypts the backup key. During the restore process, master key, which is stored in HSM, decrypts the backup key. Then, the backup key is used to restore backup contents. For information about HSM-based encryption, see HSM-based encryption for backups. 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. Go to the appropriate page or directory.

   Graphical user interface

   a. Log on to the graphical user interface.

   b. On the Welcome page, click **Administration** > **Backup and Restore**.

   Command-line interface

   a. Go to the `WAS_HOME/bin` directory. For example,

   Windows

   ```
   cd drive:\Program Files (x86)\IBM\WebSphere\AppServer\
   bin
   ```

   Linux

   ```
   cd /opt/IBM/WebSphere/AppServer/bin
   ```

   b. Start the **wsadmin** interface by using an authorized user ID, such as `SKLMAdmin`. For example,

   Windows

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

   Linux

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

   REST interface

   Open a REST client.

2. Create a backup file. You can run only one backup or restore task at a time.

   Graphical user interface

   a. On the **Backup and Restore** table, the **Backup repository location** field displays the default <*SKLM_DATA*> directory path, where the backup file is saved, for example, `C:\Program Files\IBM\WebSphere\AppServer\products\sklm\data`. For the definition of <*SKLM_DATA*>, see Definitions for *HOME* and other directory variables. Click **Browse** to specify a backup repository location under <*SKLM_DATA*> directory.

   Directory path in the **Backup repository location** field changes based on the value that you set for the **tklm.backup.dir** property in the `SKLMConfig.properties` file.

   b. Click **Create Backup**.

   c. On the Create Backup page, specify a description. A read-only backup file location is displayed in the **Backup location** field.

   d. Click **Create Backup**.

   Command-line interface

   Type **tklmBackupRun**, the backup location, and any other necessary information to create a backup file. For example:

```
print AdminTask.tklmBackupRun
 ('[-backupDirectory C:\\sklmbackup1]')
```

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 run **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:<port>/SKLM/rest/v1/ckms/backups
Content-Type: application/json
Accept : application/json
Authorization: SKLMAuth authId=139aeh34567m
Accept-Language : en
{"backupDirectory":"/sklmbackup1"}
```

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_v3.0.1.0_20170123144220-**0800**_backup.jar, where -0800 indicates that the timezone is eight hours behind GMT.

What to do next

Do not edit a file in the backup JAR file. The file that you attempt to edit becomes unreadable. Master key that was used for the backup key encryption must be intact to restore the backup file. If the master key is refreshed, all the older backups are inaccessible or unusable.

Backing up large amount of data

You must set the **enableHighScaleBackup=true** property in the SKLMConfig.properties configuration file to back up large number of encryption keys.

About this task

You can use the Backup and Restore page to back up data. Alternatively, you can use the **tklmBackupRun** command or **Backup Run REST Service**. Your role must have a permission to back up files.

Note:

- You cannot create a cross-platform compatible backup file if IBM Security Key Lifecycle Manager is configured for high performance backup and restore activities. You can use the backup file to restore data in an identical operating environment. The operating system, middleware components, and directory structures must be identical on both systems.
- The db2restore.log file is created during restore process only when IBM Security Key Lifecycle Manager is configured for high performance backup and restore operations.

Procedure

1. Set the **enableHighScaleBackup=true** property in the *<SKLM_HOME>*/config/SKLMConfig.properties file.

   Command-line interface

   a. Go to the *<WAS_HOME>*/bin directory. For example,

      Windows
      ```
      cd drive:\Program Files (x86)\IBM\WebSphere\AppServer\
      bin
      ```

      Linux `cd /opt/IBM/WebSphere/AppServer/bin`

   b. Start the **wsadmin** interface by using an authorized user ID, such as SKLMAdmin. For example,

      Windows
      ```
      wsadmin.bat -username SKLMAdmin -password mypwd -lang jython
      ```

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

   c. Run the **tklmConfigUpdateEntry** command to set **enableHighScaleBackup** property in the SKLMConfig.properties configuration file.

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

   REST interface

   a. Open a REST client.

   b. 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.

   c. Run **Update Config Property REST Service** to set **enableHighScaleBackup** property in the SKLMConfig.properties configuration file. Pass the user authentication identifier that you obtained in Step b along with the request message as shown in the following example.

      ```
      PUT https://localhost:<port>/SKLM/rest/v1/configProperties
      Content-Type: application/json
      Accept : application/json
      Authorization: SKLMAuth userAuthId=139aeh34567m
      Accept-Language : en
      { "enableHighScaleBackup" : "true"}
      ```

2. Go to the appropriate page or directory for backing up data.

   Graphical user interface

   a. Log on to the graphical user interface.

   b. On the Welcome page, click **Administration** > **Backup and Restore**.

   Command-line interface

   a. Go to the *<WAS_HOME>*/bin directory.

   b. Start the **wsadmin** interface by using an authorized user ID, such as SKLMAdmin.

   REST interface

   Open a REST client.

3. 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, the **Backup repository location** field displays the default <*SKLM_DATA*> directory path, where the backup file is saved, for example, `C:\Program Files\IBM\WebSphere\AppServer\products\sklm\data`. For the definition of <*SKLM_DATA*>, see Definitions for *HOME* and other directory variables. Click **Browse** to specify a backup repository location under <*SKLM_DATA*> directory.

   Directory path in the **Backup repository location** field changes based on the value that you set for the **`tklm.backup.dir`** property in the `SKLMConfig.properties` file.

b. Click **Create Backup**.

c. On the Create Backup page, specify information such as a value for the encryption password and backup description. A read-only backup file location is displayed in the **Backup location** field. Ensure that you retain the encryption password for future use in case you restore the backup.

   Note: If HSM-based encryption is used for the backups, you need not specify the password.

d. Click **Create Backup**.

Command-line interface

Type `tklmBackupRun` and specify the needed values to create a backup file as shown in the following example.

```
print AdminTask.tklmBackupRun ('[-backupDirectory C:\\sklmbackup1 -password myBackupPw
```

REST interface

Run **`Backup Run REST Service`** by sending the HTTP POST request as shown in the following example.

```
POST https://localhost:<port>/SKLM/rest/v1/ckms/backups
Content-Type: application/json
Accept : application/json
Authorization: SKLMAuth authId=139aeh34567m
Accept-Language : en
{"backupDirectory":"/sklmbackup1","password":"myBackupPwd"}
```

4. A message is displayed to indicate 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_v3.0.1.0_20170123144220`**`-0800`**`_backup.jar`, where -0800 indicates that the timezone is eight hours behind GMT.

   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.

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

A restore returns the IBM Security Key Lifecycle Manager server to a known state, by using backed-up production data, such as the IBM Security Key Lifecycle Manager key materials and other critical information.

Before you begin

Consider the following guidelines before you restore HSM-based encryption backups:

- Ensure that the same HSM partition is present with all its key entries intact on the system where the backup file is restored.
- Master key that was used for the backup key encryption must be intact to restore the backup file. If the master key is refreshed, all the older backups are inaccessible or unusable.
- You must connect to the same HSM and the master key for backup and restore operations irrespective of whether you use HSM-based encryption or password-based encryption.

When you run backup operation, the manifest file is created along with the backup archive. Before you restore the backup files, ensure that the backup manifest file lists all the IBM Security Key Lifecycle Manager data files in the archive.

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. IBM Security Key Lifecycle Manager server automatically restarts after the restore process is complete. Verify the environment before you bring the IBM Security Key Lifecycle Manager server back into production.

Procedure

1. Go to the appropriate page or directory:

   Graphical user interface

   a. Log on to the graphical user interface.

   b. On the Welcome page, click **Administration** > **Backup and Restore**.

   Command-line interface

   a. Go to the `WAS_HOME/bin` directory. For example,

      Windows

      ```
      cd drive:\Program Files\IBM\WebSphere\AppServer\bin
      ```

      Linux `cd /opt/IBM/WebSphere/AppServer/bin`

   b. Start the **`wsadmin`** interface by using an authorized user ID, such as `SKLMAdmin`. For example,

Windows

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

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. On the **Backup and Restore** table, select a backup file that is listed in the table.

b. 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.

c. On the Restore Backup page, specify the encryption password that was used to create the backup file.

Note: If HSM-based encryption is used for the backups, you need not specify the password.

d. Click **Restore Backup**.

Command-line interface

Type `tklmBackupRunRestore` and specify the necessary 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_v3.0.1.0_20170705235417-1200_backup
  -password myBackupPwd]')
```

Note: If HSM-based encryption is used for the backups, you need not specify the password.

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 run **`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:<port>/SKLM/rest/v1/ckms/restore
Content-Type: application/json
Accept : application/json
Authorization: SKLMAuth authId=139aeh34567m
Accept-Language : en
{"backupFilePath":"/opt/mysklmbackups/sklm_v2.7.0.0_20160705235417-1200_
backup.jar","password":"myBackupPwd"}
```

Note: If HSM-based encryption is used for the backups, you need not specify the password.

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

Note: After data restoration, ensure that the path for the properties in the `SKLMConfig.properties`, `datastore.properties`, and `ReplicationSKLMConfig.properties` files are correct before you proceed with your next task.

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.

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 **Administration** > **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 can 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. Go to the appropriate page or directory.
   - Command-line interface
     a. Go to the *<WAS_HOME>*/bin directory. For example,

        Windows
        ```
        cd drive:\Program Files\IBM\WebSphere\AppServer\bin
        ```

        Linux
        ```
        cd /opt/IBM/WebSphere/AppServer/bin
        ```

     b. Start the **wsadmin** interface by using an authorized user ID, such as SKLMAdmin. For example,

        Windows
        ```
        wsadmin.bat -username SKLMAdmin -password mypwd -lang jython
        ```

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

   - REST interface:
     - Open a REST client.
2. Complete the task.
   - Command-line interface:

     tklmBackupGetProgress

     Type tklmBackupGetProgress to determine the current phase of a backup task that is running. For example, type:
     ```
     print AdminTask.tklmBackupGetProgress()
     ```

     tklmBackupGetRestoreProgress

     Type tklmBackupGetRestoreProgress 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:\\sklmbackup1 -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:<port>/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:<port>/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:<port>/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:<port>/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:<port>/SKLM/rest/v1/ckms/backups?backupDirectory=
/sklmbackup
Content-Type: application/json
Accept : application/json
Authorization: SKLMAuth authId=139aeh34567m
Accept-Language : en
```

Backup and restore operations for earlier versions of IBM Security Key Lifecycle Manager and IBM Tivoli 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 and IBM Tivoli 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.

Note: For greater security, change the IBM Security Key Lifecycle Manager User password soon after the data migration process.

Cross-migrating IBM Tivoli Key Lifecycle Manager, Version 1.0 data

Use the IBM Security Key Lifecycle Manager backup and restore utilities to cross-migrate IBM Tivoli Key Lifecycle Manager, Version 1.0 data.

About this task

Cross-migration of IBM Tivoli Key Lifecycle Manager, Version 1.0 data to IBM Security Key Lifecycle Manager, Version 3.0.1consists of the following two stages:

1. Migrating IBM Tivoli Key Lifecycle Manager, Version 1.0 data to a system where IBM Security Key Lifecycle Manager, Version 2.7 is installed by following steps that are described in the IBM Security Key Lifecycle Manager, Version 2.7 documentation.

   Backing up IBM Tivoli Key Lifecycle Manager, Version 1.0 data

   Restoring IBM Tivoli Key Lifecycle Manager Version 1.0 backup files

2. Migrating IBM Security Key Lifecycle Manager, Version 2.7 data on a system where IBM Security Key Lifecycle Manager, Version 3.0.1 is installed by using steps that are described in the following topics.

   "Backing up IBM Security Key Lifecycle Manager, Version 2.7 data" on page 148

   "Restoring IBM Security Key Lifecycle Manager, Version 2.7 backup files" on page 150

Cross-migrating IBM Tivoli Key Lifecycle Manager, Version 2.0 data

Use the IBM Security Key Lifecycle Manager backup and restore utilities to cross-migrate IBM Tivoli Key Lifecycle Manager, Version 2.0 data.

About this task

Cross-migration of IBM Tivoli Key Lifecycle Manager, Version 2.0 data to IBM Security Key Lifecycle Manager, Version 3.0.1 consists of the following two stages:

1. Migrating IBM Tivoli Key Lifecycle Manager, Version 2.0 data to a system where IBM Security Key Lifecycle Manager, Version 2.7 is installed by following steps that are described in the IBM Security Key Lifecycle Manager, Version 2.7 documentation.

   Backing up IBM Tivoli Key Lifecycle Manager, Version 2.0 data

   Restoring IBM Tivoli Key Lifecycle Manager Version 2.0 backup files
2. Migrating IBM Security Key Lifecycle Manager, Version 2.7 data on a system where IBM Security Key Lifecycle Manager, Version 3.0.1 is installed by using steps that are described in the following topics.

   "Backing up IBM Security Key Lifecycle Manager, Version 2.7 data" on page 148

   "Restoring IBM Security Key Lifecycle Manager, Version 2.7 backup files" on page 150

Cross-migrating IBM Tivoli Key Lifecycle Manager, Version 2.0.1 data

Use the IBM Security Key Lifecycle Manager backup and restore utilities to cross-migrate IBM Tivoli Key Lifecycle Manager, Version 2.0.1 data.

About this task

Cross-migration of IBM Tivoli Key Lifecycle Manager, Version 2.0.1 data to IBM Security Key Lifecycle Manager, Version 3.0.1 consists of the following two stages:

1. Migrating IBM Tivoli Key Lifecycle Manager, Version 2.0.1 data to a system where IBM Security Key Lifecycle Manager, Version 2.7 is installed by following steps that are described in the IBM Security Key Lifecycle Manager, Version 2.7 documentation.

   Backing up IBM Tivoli Key Lifecycle Manager, Version 2.0.1 data

   Restoring IBM Tivoli Key Lifecycle Manager Version 2.0.1 backup files
2. Migrating IBM Security Key Lifecycle Manager, Version 2.7 data on a system where IBM Security Key Lifecycle Manager, Version 3.0.1 is installed by using steps that are described in the following topics.

   "Backing up IBM Security Key Lifecycle Manager, Version 2.7 data" on page 148

   "Restoring IBM Security Key Lifecycle Manager, Version 2.7 backup files" on page 150

Backing up Encryption Key Manager, Version 2.1 data

Use the IBM Security Key Lifecycle Manager, Version 3.0.1 backup utility to create Encryption Key Manager, Version 2.1 backup files.

Before you begin

- You must install IBM Security Key Lifecycle Manager, Version 3.0.1 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 3.0.1 across operating systems.

Note: For greater security, change the IBM Security Key Lifecycle Manager User password soon after the data migration process.

Procedure

1. Copy the Encryption Key Manager folder and all other necessary files to a system where IBM Security Key Lifecycle Manager, Version 3.0.1 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 3.0.1 is installed.

   Windows

   <SKLM_INSTALL_HOME>\migration\utilities\ekm21

   Default location is C:\Program Files\IBM\SKLMV301\migration\utilities\ekm21.

   Linux

   <SKLM_INSTALL_HOME>/migration/utilities/ekm21

   Default location is /opt/IBM/SKLMV301/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\\IBM\\WebSphere\\AppServer\\java\8.0
   ```

   Linux

   ```
   KLM_VERSION=2.1
   BACKUP_DIR=/ekm_backup
   EKM_HOME=/EKM21
   BACKUP_PASSWORD=passw0rd123
   JAVA_HOME=/opt/IBM/WebSphere/AppServer/java/8.0
   ```

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 3.0.1 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 3.0.1 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: You must have IBM Security Key Lifecycle Manager User role to run the 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 3.0.1 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. IBM Security Key Lifecycle Manager server automatically restarts after the restore process is complete. Verify the environment before you bring the IBM Security Key Lifecycle Manager server back into production.

Note: For greater security, change the IBM Security Key Lifecycle Manager User password soon after the data migration process.

Procedure

1. Log on to the system where IBM Security Key Lifecycle Manager, Version 3.0.1 is installed.
2. Copy the backup file, for example `sklm_vEKM21_20170420113253+0530_backup.jar`, from Encryption Key Manager, Version 2.1 system to a folder of your choice under *<SKLM_DATA>* directory, for example, `C:\Program Files\IBM\WebSphere\AppServer\products\sklm\data`. For the definition of *<SKLM_DATA>*, see Definitions for *HOME* and other directory variables.
3. Restore the backup file by using any of the following methods.

Graphical user interface	1. Log on to the graphical user interface as an authorized user, for example, `SKLMAdmin`. 2. On the Welcome page, click **Administration** > **Backup and Restore**. 3. Click **Browse** to specify the Encryption Key Manager backup file location under *<SKLM_DATA>* directory. 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	1. Go to the `<WAS_HOME>/bin` directory. For example,
	Windows
	`cd drive:\Program Files\IBM\WebSphere\AppServer\bin`
	Linux `cd /opt/IBM/WebSphere/AppServer/bin`
	2. Start the **`wsadmin`** interface by using an authorized user ID, such as `SKLMAdmin`. For example,
	Windows
	`wsadmin.bat -username SKLMAdmin -password mypwd -lang jython`
	Linux
	`./wsadmin.sh -username SKLMAdmin -password mypwd -lang jython`
	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.
	`print AdminTask.tklmBackupRunRestore`
	`('[-backupFilePath <SKLM_DATA>/sklm_vEKM21_20170420113253+0530_backup.ja`
	`-password myBackupPwd]')`
	4. Restart IBM Security Key Lifecycle Manager server.

REST interface	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:<port>/SKLM/rest/v1/ckms/restore`
	`Content-Type: application/json`
	`Accept: application/json`
	`Authorization: SKLMAuth authId=139aeh34567m`
	`Accept-Language: en`
	`{"backupFilePath":"<SKLM_DATA>/sklm_vEKM21_20170420113253+0530_backup.jar`
	`backup.jar","password":"myBackupPwd"}`
	4. Restart IBM Security Key Lifecycle Manager server.

Migration restore script	1. Locate the IBM Security Key Lifecycle Manager restore utilities.
	Windows
	`<SKLM_INSTALL_HOME>\migration\utilities\ekm21`
	Default location is `C:\Program Files\IBM\SKLMV30\migration\utilities\ekm21`.
	Linux `<SKLM_INSTALL_HOME>/migration/utilities/ekm21`
	Default location is `/opt/IBM/SKLMV301/migration/utilities/ekm21`.
	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.
	Window
	`WAS_HOME=C:\\Program Files\\IBM\\WebSphere\\AppServer`
	`JAVA_HOME=C:\\Program Files\\IBM\\WebSphere\\AppServer\\java\8.0`
	`BACKUP_PASSWORD=passw0rd123`
	`DB_PASSWORD=db2_password`
	`RESTORE_FILE=<SKLM_DATA>\\sklm_vEKM21_20170424024117-0400_backup.`
	`WAS_USER_PWD=wasadmin_password`
	`RESTORE_USER_ROLES=n`
	Linux
	`WAS_HOME=/opt/IBM/WebSphere/AppServer`
	`JAVA_HOME=/opt/IBM/WebSphere/AppServer/java/8.0`
	`BACKUP_PASSWORD=passw0rd123`
	`DB_PASSWORD=db2_password`
	`RESTORE_FILE=<SKLM_DATA>/20170424024117-0400_backup.jar`
	`WAS_USER_PWD=wasadmin_password`
	`RESTORE_USER_ROLES=n`
	To log in to IBM Security Key Lifecycle Manager by using the user credentials that are specified during product installation, set the **`RESTORE_USER_ROLES`** property set as “n ”. Setting the property to “n ” ensures that user ID and the password are not overwritten with the user credentials of the older version.
	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.
	`C:\\ekm_restore`
	Or
	`C:/ekm_restore`
	3. Open a command prompt and run the restore utility.
	Windows
	Go to the `<SKLM_INSTALL_HOME>\migration\utilities\ekm21` directory and run the following command:
	`restoreEKM21.bat`
	Linux Go to the `<SKLM_INSTALL_HOME>/migration/utilities/ekm21` directory and run the following command:
	`restoreEKM21.sh`
	4. Restart IBM Security Key Lifecycle Manager server.

What to do next

Note: After data restoration, ensure that the path for the properties in the `SKLMConfig.properties`, `datastore.properties`, and `ReplicationSKLMConfig.properties` files are correct before you proceed with your next task.

Backing up IBM Security Key Lifecycle Manager, Version 2.5 data

Use the IBM Security Key Lifecycle Manager, Version 3.0.1 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 3.0.1 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 3.0.1 across operating systems.

Note: For greater security, change the IBM Security Key Lifecycle Manager User password soon after the data migration process.

Procedure

Run the followings steps on systems where the IBM Security Key Lifecycle Manager version 3.0.1 and version 2.5 are installed.

IBM Security Key Lifecycle Manager, Version3.0.1	1. Log on to the system with your user credentials. 2. Locate the backup utilities folder. **Windows** `<`*`SKLM_INSTALL_HOME`*`>\migration\utilities\sklmv25` Default location is `C:\Program Files\IBM\.SKLMV301\migration\utilities\sklmv25` **Linux** `<`*`SKLM_INSTALL_HOME`*`>/migration/utilities/sklmv25` Default location is `/opt/IBM/SKLMV301/migration/utilities/sklmv25`.

IBM Security Key Lifecycle Manager, Version 2.5	1. Log on to the system with your user credentials.
	2. Copy `sklmv25` folder from the system where IBM Security Key Lifecycle Manager, Version 3.0.1 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=sklmdb2`
	`WAS_USER_PWD=wasadmin`
	`BACKUP_DIR=C:\\sklmv25_backup`
	Linux
	`WAS_HOME=/opt/IBM/WebSphere/AppServer`
	`BACKUP_PASSWORD=passw0rd123`
	`DB_PASSWORD=sklmdb2`
	`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 3.0.1 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 3.0.1 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: You must have IBM Security Key Lifecycle Manager User role to run the 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 3.0.1 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. IBM Security Key Lifecycle Manager server automatically restarts after the restore process is complete. Verify the environment before you bring the IBM Security Key Lifecycle Manager server back into production.

Note: For greater security, change the IBM Security Key Lifecycle Manager User password soon after the data migration process.

Procedure

1. Log on to the system where IBM Security Key Lifecycle Manager Version 3.0.1 is installed.
2. Copy the backup file from version 2.5 system to a folder of your choice under <*SKLM_DATA*> directory, for example, `C:\Program Files\IBM\WebSphere\AppServer\products\sklm\data\sklm_v2.5.0.3_20170429013250-0400_migration_backup.jar`. For the definition of <*SKLM_DATA*>, see Definitions for *HOME* and other directory variables.
3. Restore the backup file by using any of the following methods.

Graphical user interface	1. Log on to the graphical user interface as an authorized user, for example, `SKLMAdmin`. 2. On the Welcome page, click **Administration** > **Backup and Restore**. 3. Click **Browse** to specify the version 2.5 backup file location under *<SKLM_DATA>* directory. 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. **Note:** By using the graphical user interface, you cannot restore roles, users, and groups from IBM Security Key Lifecycle Manager, Version 2.5 backup.

Command-line interface	1. Go to the `<WAS_HOME>/bin` directory. For example,
	Windows
	`cd drive:\Program Files\IBM\WebSphere\AppServer\bin`
	Linux `cd /opt/IBM/WebSphere/AppServer/bin`
	2. Start the **`wsadmin`** interface by using an authorized user ID, such as `SKLMAdmin`. For example,
	Windows
	`wsadmin.bat -username SKLMAdmin -password mypwd -lang jython`
	Linux
	`./wsadmin.sh -username SKLMAdmin -password mypwd -lang jython`
	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.
	`print AdminTask.tklmBackupRunRestore`
	`('[-backupFilePath <SKLM_DATA>/sklm_v2.5.0.3_20170429013250-0400_migration_`
	`-password myBackupPwd]')`
	4. Restart IBM Security Key Lifecycle Manager server.
	Note: By using the command-line interface, you cannot restore roles, users, and groups from IBM Security Key Lifecycle Manager, Version 2.5 backup.

REST interface	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:<port>/SKLM/rest/v1/ckms/restore`
	`Content-Type: application/json`
	`Accept: application/json`
	`Authorization: SKLMAuth authId=139aeh34567m`
	`Accept-Language: en`
	`{"backupFilePath":"<`*`SKLM_DATA`*`>/sklm_v2.5.0.3_20170429013250-0400_migration_bac`
	`"password":"myBackupPwd"}`
	4. Restart IBM Security Key Lifecycle Manager server.
	Note: By using the REST interface, you cannot restore roles, users, and groups from IBM Security Key Lifecycle Manager, Version 2.5 backup.

Migration restore script

1. Locate the IBM Security Key Lifecycle Manager restore utilities.

   Windows

   <*SKLM_INSTALL_HOME*>\migration\utilities\sklmv25

   Default location is C:\Program Files\IBM\SKLMV301\migration\utilities\sklmv25.

   Linux

   <*SKLM_INSTALL_HOME*>/migration/utilities/sklmv25

   Default location is /opt/IBM/SKLMV301/migration/utilities/sklmv25.

2. Edit restore.properties in the sklmv25 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.

   Windows

   ```
   WAS_HOME=C:\\Program Files\\IBM\\WebSphere\\AppServer
   JAVA_HOME=C:\\Program Files\\IBM\\WebSphere\\AppServer\\java\8.0
   BACKUP_PASSWORD=passw0rd123
   DB_PASSWORD=db2_password
   RESTORE_FILE=<SKLM_DATA>\\sklm_v2.5.0.3_20170429013250-0400_migratio
   WAS_USER_PWD=wasadmin_password
   RESTORE_USER_ROLES=y
   #pkcs11_config=C:\\luna.cfg
   ```

   Linux

   ```
   WAS_HOME=/opt/IBM/WebSphere/AppServer
   JAVA_HOME=/opt/IBM/WebSphere/AppServer/java/8.0
   BACKUP_PASSWORD=passw0rd123
   DB_PASSWORD=db2_password
   RESTORE_FILE=<SKLM_DATA>/sklm_v2.5.0.3_20170429013250-0400_migration
   WAS_USER_PWD=wasadmin_password
   RESTORE_USER_ROLES=y
   #pkcs11_config=/luna.cfg
   ```

   Note:

   - To log on to IBM Security Key Lifecycle Manager by using the user credentials that are specified during product installation, set the **RESTORE_USER_ROLES** property set as "n ". Setting the property to "n " ensures that user ID and the password are not overwritten with the user credentials of the older version.
   - If IBM Security Key Lifecycle Manager is configured with HSM, uncomment the **#pkcs11_config** property and specify the correct path of luna.cfg file as the value.
   - On Windows operating system, when you specify path in the properties file, use either "/ " or "\\ " as path separator as shown in the following example.

     C:\\sklmv25_restore

     Or

     C:/sklmv25_restore

3. Open a command prompt and run the restore utility.

   Windows

   Go to the <*SKLM_INSTALL_HOME*>\migration\utilities\sklmv25 directory and run the following command:

   restoreV25.bat

   Linux

   Go to the <*SKLM_INSTALL_HOME*>/migration/utilities/sklmv25 directory and run the following command:

   restoreV25.sh

4. Restart IBM Security Key Lifecycle Manager server.

Note: By using the migration restore script, you can restore users, groups, and roles from IBM Security Key Lifecycle Manager, Version 2.5 backup. Ensure that the value of the **WAS_USER_PWD** parameter for

Note: If you want to set up a Multi-Master cluster on the restored IBM Security Key Lifecycle Manager server, in the `SKLMConfig.properties` file, update the **`enableClientCertPush`** property as follows:

```
enableClientCertPush=true
```

You can use the Update Config Property REST Service or the tklmConfigUpdateEntry CLI command to update the property.

Then, restart the IBM Security Key Lifecycle Manager server.

What to do next

Note: After data restoration, ensure that the path for the properties in the `SKLMConfig.properties`, `datastore.properties`, and `ReplicationSKLMConfig.properties` files are correct before you proceed with your next task.

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.

Backing up IBM Security Key Lifecycle Manager, Version 2.6 data

Use the IBM Security Key Lifecycle Manager, Version 3.0.1 backup utility to create IBM Security Key Lifecycle Manager, Version 2.6 cross-platform backup files.

Before you begin

You must install IBM Security Key Lifecycle Manager, Version 3.0.1 on a system. Ensure that the system with IBM Security Key Lifecycle Manager, Version 2.6 with fix pack 2 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 3.0.1 across operating systems.

Note: For greater security, change the IBM Security Key Lifecycle Manager User password soon after the data migration process.

Procedure

Run the followings steps on systems where the IBM Security Key Lifecycle Manager version 3.0.1 and version 2.6 are installed.

IBM Security Key Lifecycle Manager, Version3.0.1	1. Log on to the system with your user credentials. 2. Locate the backup utilities folder. **Windows** `<SKLM_INSTALL_HOME>\migration\utilities\sklmv26` Default location is `C:\Program Files\IBM\SKLMV301\migration\utilities\sklmv26`. **Linux** `<SKLM_INSTALL_HOME>/migration/utilities/sklmv26` Default location is `/opt/IBM/SKLMV301/migration/utilities/sklmv26`.

IBM Security Key Lifecycle Manager, Version 2.6	1. Log on to the system with your user credentials.
	2. Copy `sklmv26` folder from the system where IBM Security Key Lifecycle Manager, Version 3.0.1 is installed to a local directory of your choice.
	3. Edit `backup.properties` in the `sklmv26` 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=sklmdb2`
	`WAS_USER_PWD=wasadmin`
	`BACKUP_DIR=C:\\sklmv26_backup`
	Linux
	`WAS_HOME=/opt/IBM/WebSphere/AppServer`
	`BACKUP_PASSWORD=passw0rd123`
	`DB_PASSWORD=sklmdb2`
	`WAS_USER_PWD=wasadmin`
	`BACKUP_DIR=/sklmv26_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:\\sklmv26_backup`
	Or
	`C:/sklmv26_backup`
	4. Open a command prompt and run the backup utility.
	Windows
	Go to the `sklmv26` directory (see Step b) and run the following command:
	`backupV26.bat`
	Linux Go to the `sklmv26` directory (see Step b) and run the following command:
	`backupV26.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 2.6 backup files

You can restore the IBM Security Key Lifecycle Manager, Version 2.6 cross-platform backup files on a system with IBM Security Key Lifecycle Manager, Version 3.0.1 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 3.0.1 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: You must have IBM Security Key Lifecycle Manager User role to run the backup and restore operations.

About this task

You can restore IBM Security Key Lifecycle Manager, Version 2.6 cross-platform compatible backup files on a system with IBM Security Key Lifecycle Manager, Version 3.0.1 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. IBM Security Key Lifecycle Manager server automatically restarts after the restore process is complete. Verify the environment before you bring the IBM Security Key Lifecycle Manager server back into production.

Note: For greater security, change the IBM Security Key Lifecycle Manager User password soon after the data migration process.

Procedure

1. Log on to the system where IBM Security Key Lifecycle Manager, Version 3.0.1 is installed.
2. Copy the backup file from version 2.6 system to a folder of your choice under *<SKLM_DATA>* directory, for example, `C:\Program Files\IBM\WebSphere\AppServer\products\sklm\data\sklm_v2.6.0.2_20170429013250-0400_migration_backup.jar`. For the definition of *<SKLM_DATA>*, see Definitions for *HOME* and other directory variables.
3. Restore the backup file by using any of the following methods.

Graphical user interface	1. Log on to the graphical user interface as an authorized user, for example, `SKLMAdmin`. 2. On the Welcome page, click **Administration** > **Backup and Restore**. 3. Click **Browse** to specify the version 2.6 backup file location under <*SKLM_DATA*> directory. 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. **Note:** By using the graphical user interface, you cannot restore roles, users, and groups from IBM Security Key Lifecycle Manager, Version 2.6 backup.

Command-line interface	1. Go to the `*WAS_HOME*/bin` directory. For example,
	Windows
	`cd drive:\Program Files\IBM\WebSphere\AppServer\bin`
	Linux `cd /opt/IBM/WebSphere/AppServer/bin`
	2. Start the **`wsadmin`** interface by using an authorized user ID, such as `SKLMAdmin`. For example,
	Windows
	`wsadmin.bat -username SKLMAdmin -password mypwd -lang jython`
	Linux
	`./wsadmin.sh -username SKLMAdmin -password mypwd -lang jython`
	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.
	`print AdminTask.tklmBackupRunRestore`
	`('[-backupFilePath <SKLM_DATA>/sklm_v2.6.0.2_20170429013250-0400_migration_`
	`-password myBackupPwd]')`
	4. Restart IBM Security Key Lifecycle Manager server.
	Note: By using the command-line interface, you cannot restore roles, users, and groups from IBM Security Key Lifecycle Manager, Version 2.6 backup.

REST interface	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:<port>/SKLM/rest/v1/ckms/restore`
	`Content-Type: application/json`
	`Accept: application/json`
	`Authorization: SKLMAuth authId=139aeh34567m`
	`Accept-Language: en`
	`{"backupFilePath":"<`*`SKLM_DATA`*`>/sklm_v2.6.0.2_20170429013250-0400_migration_bac`
	`"password":"myBackupPwd"}`
	4. Restart IBM Security Key Lifecycle Manager server.
	Note: By using the REST interface, you cannot restore roles, users, and groups from IBM Security Key Lifecycle Manager, Version 2.6 backup.

Migration restore script

1. Locate the IBM Security Key Lifecycle Manager restore utilities.

   Windows

   <*SKLM_INSTALL_HOME*>\migration\utilities\sklmv26

   Default location is C:\Program Files\IBM\SKLMV301\migration\utilities\sklmv26.

   Linux

   <*SKLM_INSTALL_HOME*>/migration/utilities/sklmv26

   Default location is /opt/IBM/SKLMV301/migration/utilities/sklmv26.

2. Edit restore.properties in the sklmv26 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.

   Windows

   ```
   WAS_HOME=C:\\Program Files\\IBM\\WebSphere\\AppServer
   JAVA_HOME=C:\\Program Files\\IBM\\WebSphere\\AppServer\\java\8.0
   BACKUP_PASSWORD=passw0rd123
   DB_PASSWORD=db2_password
   RESTORE_FILE=<SKLM_DATA>\\sklm_v2.6.0.2_20170429013250-0400_migratio
   WAS_USER_PWD=wasadmin_password
   RESTORE_USER_ROLES=y
   #pkcs11_config=C:\\luna.cfg
   ```

   Linux

   ```
   WAS_HOME=/opt/IBM/WebSphere/AppServer
   JAVA_HOME=/opt/IBM/WebSphere/AppServer/java/8.0
   BACKUP_PASSWORD=passw0rd123
   DB_PASSWORD=db2_password
   RESTORE_FILE=<SKLM_DATA>/sklm_v2.6.0.2_20170429013250-0400_migration
   WAS_USER_PWD=wasadmin_password
   RESTORE_USER_ROLES=y
   #pkcs11_config=/luna.cfg
   ```

   Note:

   - To log on to IBM Security Key Lifecycle Manager by using the user credentials that are specified during product installation, set the **RESTORE_USER_ROLES** property set as "n ". Setting the property to "n " ensures that user ID and the password are not overwritten with the user credentials of the older version.

     RESTORE_USER_ROLES=n
   - If IBM Security Key Lifecycle Manager is configured with HSM, uncomment the **#pkcs11_config** property and specify the correct path of luna.cfg file as the value.
   - On Windows operating system, when you specify path in the properties file, use either "/ " or "\\ " as path separator as shown in the following example.

     C:\\sklmv26_restore

     Or

     C:/sklmv26_restore

3. Open a command prompt and run the restore utility.

   Windows

   Go to the <*SKLM_INSTALL_HOME*>\migration\utilities\sklmv26 directory and run the following command:

   restoreV26.bat

   Linux

   Go to the <*SKLM_INSTALL_HOME*>/migration/utilities/sklmv26 directory and run the following command:

   restoreV26.sh

4. Restart IBM Security Key Lifecycle Manager server.

Note: By using the migration restore script, you can restore users, groups, and roles from IBM Security Key Lifecycle Manager, Version

What to do next

Note: After data restoration, ensure that the path for the properties in the `SKLMConfig.properties`, `datastore.properties`, and `ReplicationSKLMConfig.properties` files are correct before you proceed with your next task.

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 160.

Backing up IBM Security Key Lifecycle Manager, Version 2.7 data

Use the IBM Security Key Lifecycle Manager, Version 3.0.1 backup utility to create IBM Security Key Lifecycle Manager, Version 2.7 cross-platform backup files.

Before you begin

You must install IBM Security Key Lifecycle Manager, Version 3.0.1 on a system. Ensure that the system with IBM Security Key Lifecycle Manager, Version 2.7 General Availability (GA) 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 3.0.1 across operating systems.

Note: For greater security, change the IBM Security Key Lifecycle Manager User password soon after the data migration process.

Procedure

Run the followings steps on systems where the IBM Security Key Lifecycle Manager version 3.0.1 and version 2.7 are installed.

IBM Security Key Lifecycle Manager, Version 3.0.1	1. Log on to the system with your user credentials. 2. Locate the backup utilities folder. **Windows** `<SKLM_INSTALL_HOME>\migration\utilities\sklmv27` Default location is `C:\Program Files\IBM\SKLMV301\migration\utilities\sklmv27`. **Linux** `<SKLM_INSTALL_HOME>/migration/utilities/sklmv27` Default location is `/opt/IBM/SKLMV301/migration/utilities/sklmv27`.

IBM Security Key Lifecycle Manager, Version 2.7	1. Log on to the system with your user credentials.
	2. Copy `sklmv27` folder from the system where IBM Security Key Lifecycle Manager, Version 3.0.1 is installed to a local directory of your choice.
	3. Edit `backup.properties` in the `sklmv27` 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\\IBM\\WebSphere\\AppServer`
	`JAVA_HOME=C:\\Program Files\\IBM\\WebSphere\\AppServer\\java\\8.0 (ma`
	`BACKUP_PASSWORD=passw0rd123`
	`DB_PASSWORD=sklmdb2`
	`WAS_USER_PWD=wasadmin`
	`MIGRATE_INSTANCE_ID=NO (optional)`
	Linux
	`WAS_HOME=/opt/IBM/WebSphere/AppServer`
	`JAVA_HOME=/opt/IBM/WebSphere/AppServer/java/8.0`
	`BACKUP_PASSWORD=passw0rd123`
	`DB_PASSWORD=sklmdb2`
	`WAS_USER_PWD=wasadmin`
	`MIGRATE_INSTANCE_ID=NO (optional)`
	Specify `MIGRATE_INSTANCE_ID=YES` to migrate the IBM Security Key Lifecycle Manager instance ID. If you do not specify this property, the instance ID is not migrated to IBM Security Key Lifecycle Manager, Version 3.0.1.
	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:\\sklmv27_backup`
	Or
	`C:/sklmv27_backup`
	4. Open a command prompt and run the backup utility.
	Windows
	Go to the `sklmv27` directory (see Step b) and run the following command:
	`backupV27.bat`
	Linux Go to the `sklmv27` directory (see Step b) and run the following command:
	`backupV27.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 2.7 backup files

You can restore the IBM Security Key Lifecycle Manager, Version 2.7 cross-platform backup files on a system with IBM Security Key Lifecycle Manager, Version 3.0.1 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 3.0.1 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: You must have IBM Security Key Lifecycle Manager User role to run the backup and restore operations.

About this task

You can restore IBM Security Key Lifecycle Manager, Version 2.7 cross-platform compatible backup files on a system with IBM Security Key Lifecycle Manager, Version 3.0.1 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. IBM Security Key Lifecycle Manager server automatically restarts after the restore process is complete. Verify the environment before you bring the IBM Security Key Lifecycle Manager server back into production.

Note: For greater security, change the IBM Security Key Lifecycle Manager User password soon after the data migration process.

Procedure

1. Log on to the system where IBM Security Key Lifecycle Manager, Version 3.0.1 is installed.
2. Copy the backup file from version 2.7 system to a folder of your choice under <*SKLM_DATA*> directory, for example, `C:\Program Files\IBM\WebSphere\AppServer\products\sklm\data\sklm_v2.7.0.0_20170429013250-0400_migration_backup.jar`. For the definition of <*SKLM_DATA*>, see Definitions for *HOME* and other directory variables.
3. Restore the backup file by using any of the following methods.

Graphical user interface	1. Log on to the graphical user interface as an authorized user, for example, `SKLMAdmin`. 2. On the Welcome page, click **Administration** > **Backup and Restore**. 3. Click **Browse** to specify the version 2.7 backup file location under <*SKLM_DATA*> directory. 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. **Note:** By using the graphical user interface, you cannot restore roles, users, and groups from IBM Security Key Lifecycle Manager, Version 2.7 backup.

Command-line interface	1. Go to the `WAS_HOME/bin` directory. For example,
	Windows
	`cd drive:\Program Files\IBM\WebSphere\AppServer\bin`
	Linux `cd /opt/IBM/WebSphere/AppServer/bin`
	2. Start the **`wsadmin`** interface by using an authorized user ID, such as `SKLMAdmin`. For example,
	Windows
	`wsadmin.bat -username SKLMAdmin -password mypwd -lang jython`
	Linux
	`./wsadmin.sh -username SKLMAdmin -password mypwd -lang jython`
	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.
	`print AdminTask.tklmBackupRunRestore`
	`('[-backupFilePath <SKLM_DATA>/sklm_v2.7.0.0_20170429013250-0400_migration_`
	`-password myBackupPwd]')`
	4. Restart IBM Security Key Lifecycle Manager server.
	Note: By using the command-line interface, you cannot restore roles, users, and groups from IBM Security Key Lifecycle Manager, Version 2.7 backup.

<table>
<tr>
<td><b>REST interface</b></td>
<td>
<ol>
<li>Open a REST client.</li>
<li>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.</li>
<li>To invoke <b><code>Backup Run Restore REST Service</code></b>, 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 <code>Step b</code> along with the request message as shown in the following example.
<pre>POST https://localhost:<port>/SKLM/rest/v1/ckms/restore
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth authId=139aeh34567m
Accept-Language: en
{"backupFilePath":"&lt;<i>SKLM_DATA</i>&gt;/sklm_v2.7.0.0_20170429013250-0400_migration_bac
"password":"myBackupPwd"}</pre>
</li>
<li>Restart IBM Security Key Lifecycle Manager server.</li>
</ol>
<b>Note:</b> By using the REST interface, you cannot restore roles, users, and groups from IBM Security Key Lifecycle Manager, Version 2.7 backup.
</td>
</tr>
</table>

Migration restore script

1. Locate the IBM Security Key Lifecycle Manager restore utilities.

   Windows
   `<SKLM_INSTALL_HOME>\migration\utilities\sklmv27`

   Default location is `C:\Program Files\IBM\SKLMV301\migration\utilities\sklmv27`.

   Linux `<SKLM_INSTALL_HOME>/migration/utilities/sklmv27`

   Default location is `/opt/IBM/SKLMV301/migration/utilities/sklmv27`.
2. Edit `restore.properties` in the `sklmv27` 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.

   Windows
   ```
   WAS_HOME=C:\\Program Files\\IBM\\WebSphere\\AppServer
   JAVA_HOME=C:\\Program Files\\IBM\\WebSphere\\AppServer\\java\8.0
   BACKUP_PASSWORD=passw0rd123
   DB_PASSWORD=db2_password
   RESTORE_FILE=<SKLM_DATA>\\sklm_v2.7.0.0_20170429013250-0400_migratio
   WAS_USER_PWD=wasadmin_password
   RESTORE_USER_ROLES=y
   #pkcs11_config=C:\\luna.cfg
   ```

   Linux
   ```
   WAS_HOME=/opt/IBM/WebSphere/AppServer
   JAVA_HOME=/opt/IBM/WebSphere/AppServer/java/8.0
   BACKUP_PASSWORD=passw0rd123
   DB_PASSWORD=db2_password
   RESTORE_FILE=<SKLM_DATA>/sklm_v2.7.0.0_20170429013250-0400_migration
   WAS_USER_PWD=wasadmin_password
   RESTORE_USER_ROLES=y
   #pkcs11_config=/luna.cfg
   ```
   Note:
   - To log on to IBM Security Key Lifecycle Manager by using the user credentials that are specified during product installation, set the **RESTORE_USER_ROLES** property set as "n ". Setting the property to "n " ensures that user ID and the password are not overwritten with the user credentials of the older version.

     `RESTORE_USER_ROLES=n`
   - If IBM Security Key Lifecycle Manager is configured with HSM, uncomment the **#pkcs11_config** property and specify the correct path of `luna.cfg` file as the value.
   - On Windows operating system, when you specify path in the properties file, use either "/ " or "\\ " as path separator as shown in the following example.

     `C:\\sklmv27_restore`

     Or

     `C:/sklmv27_restore`
3. Open a command prompt and run the restore utility.

   Windows
   Go to the `<SKLM_INSTALL_HOME>\migration\utilities\sklmv27` directory and run the following command:

   `restoreV27.bat`

   Linux Go to the `<SKLM_INSTALL_HOME>/migration/utilities/sklmv27` directory and run the following command:

   `restoreV27.sh`
4. Restart IBM Security Key Lifecycle Manager server.

Note: By using the migration restore script, you can restore users, groups, and roles from IBM Security Key Lifecycle Manager, Version

What to do next

Note: After data restoration, ensure that the path for the properties in the `SKLMConfig.properties`, `datastore.properties`, and `ReplicationSKLMConfig.properties` files are correct before you proceed with your next task.

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 160.

Backing up IBM Security Key Lifecycle Manager, Version 3.0 data

Use the IBM Security Key Lifecycle Manager, Version 3.0.1 backup utility to create IBM Security Key Lifecycle Manager, Version 3.0 cross-platform backup files.

Before you begin

You must install IBM Security Key Lifecycle Manager, Version 3.0.1 on a system. Ensure that the system with IBM Security Key Lifecycle Manager, Version 3.0 General Availability (GA) 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 3.0.1 across operating systems.

Note: For greater security, change the IBM Security Key Lifecycle Manager User password soon after the data migration process.

Procedure

Run the followings steps on systems where the IBM Security Key Lifecycle Manager version 3.0.1 and version 3.0 are installed.

IBM Security Key Lifecycle Manager, Version 3.0.1	1. Log on to the system with your user credentials. 2. Locate the backup utilities folder. **Windows** `<SKLM_INSTALL_HOME>\migration\utilities\sklmv30` Default location is `C:\Program Files\IBM\SKLMV301\migration\utilities\sklmv30`. **Linux** `<SKLM_INSTALL_HOME>/migration/utilities/sklmv30` Default location is `/opt/IBM/SKLMV301/migration/utilities/sklmv30`.

IBM Security Key Lifecycle Manager, Version 3.0	1. Log on to the system with your user credentials.
	2. Copy `sklmv30` folder from the system where IBM Security Key Lifecycle Manager, Version 3.0.1 is installed to a local directory of your choice.
	3. Edit `backup.properties` in the `sklmv30` 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\\IBM\\WebSphere\\AppServer`
	`JAVA_HOME=C:\\Program Files\\IBM\\WebSphere\\AppServer\\java\\8.0 (m`
	`BACKUP_PASSWORD=passw0rd123`
	`DB_PASSWORD=sklmdb2`
	`WAS_USER_PWD=wasadmin`
	`MIGRATE_INSTANCE_ID=NO (optional)`
	Linux
	`WAS_HOME=/opt/IBM/WebSphere/AppServer`
	`JAVA_HOME=/opt/IBM/WebSphere/AppServer/java/8.0`
	`BACKUP_PASSWORD=passw0rd123`
	`DB_PASSWORD=sklmdb2`
	`WAS_USER_PWD=wasadmin`
	`MIGRATE_INSTANCE_ID=NO (optional)`
	Specify `MIGRATE_INSTANCE_ID=YES` to migrate the IBM Security Key Lifecycle Manager instance ID. If you do not specify this property, the instance ID is not migrated to IBM Security Key Lifecycle Manager, Version 3.0.1.
	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:\\sklmv30_backup`
	Or
	`C:/sklmv30_backup`
	4. Open a command prompt and run the backup utility.
	Windows
	Go to the `sklmv30` directory (see Step b) and run the following command:
	`backupV30.bat`
	Linux Go to the `sklmv30` directory (see Step b) and run the following command:
	`backupV30.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 3.0 backup files

You can restore the IBM Security Key Lifecycle Manager, Version 3.0 cross-platform backup files on a system with IBM Security Key Lifecycle Manager, Version 3.0.1 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 3.0.1 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: You must have IBM Security Key Lifecycle Manager User role to run the backup and restore operations.

About this task

You can restore IBM Security Key Lifecycle Manager, Version 3.0 cross-platform compatible backup files on a system with IBM Security Key Lifecycle Manager, Version 3.0.1 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. IBM Security Key Lifecycle Manager server automatically restarts after the restore process is complete. Verify the environment before you bring the IBM Security Key Lifecycle Manager server back into production.

Note: For greater security, change the IBM Security Key Lifecycle Manager User password soon after the data migration process.

Procedure

1. Log on to the system where IBM Security Key Lifecycle Manager, Version 3.0.1 is installed.
2. Copy the backup file from version 3.0 system to a folder of your choice under <*SKLM_DATA*> directory, for example, `C:\Program Files\IBM\WebSphere\AppServer\products\sklm\data\sklm_v3.0.0.0_20170429013250-0400_migration_backup.jar`. For the definition of <*SKLM_DATA*>, see Definitions for *HOME* and other directory variables.
3. Restore the backup file by using any of the following methods.

Graphical user interface	1. Log on to the graphical user interface as an authorized user, for example, `SKLMAdmin`. 2. On the Welcome page, click **Administration** > **Backup and Restore**. 3. Click **Browse** to specify the version 3.0 backup file location under <*SKLM_DATA*> directory. 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. **Note:** By using the graphical user interface, you cannot restore roles, users, and groups from IBM Security Key Lifecycle Manager, Version 3.0 backup.

Command-line interface	1. Go to the *`WAS_HOME`*`/bin` directory. For example,
	Windows
	`cd drive:\Program Files\IBM\WebSphere\AppServer\bin`
	Linux `cd /opt/IBM/WebSphere/AppServer/bin`
	2. Start the **`wsadmin`** interface by using an authorized user ID, such as `SKLMAdmin`. For example,
	Windows
	`wsadmin.bat -username SKLMAdmin -password mypwd -lang jython`
	Linux
	`./wsadmin.sh -username SKLMAdmin -password mypwd -lang jython`
	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.
	`print AdminTask.tklmBackupRunRestore`
	`('[-backupFilePath <SKLM_DATA>/sklm_v3.0.0.0_20170429013250-0400_migration_`
	`-password myBackupPwd]')`
	4. Restart IBM Security Key Lifecycle Manager server.
	Note: By using the command-line interface, you cannot restore roles, users, and groups from IBM Security Key Lifecycle Manager, Version 3.0 backup.

REST interface	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:<port>/SKLM/rest/v1/ckms/restore`
	`Content-Type: application/json`
	`Accept: application/json`
	`Authorization: SKLMAuth authId=139aeh34567m`
	`Accept-Language: en`
	`{"backupFilePath":"<SKLM_DATA>/sklm_v3.0.0.0_20170429013250-0400_migration_bac`
	`"password":"myBackupPwd"}`
	4. Restart IBM Security Key Lifecycle Manager server.
	Note: By using the REST interface, you cannot restore roles, users, and groups from IBM Security Key Lifecycle Manager, Version 3.0 backup.

Migration restore script

1. Locate the IBM Security Key Lifecycle Manager restore utilities.

   Windows
   `<SKLM_INSTALL_HOME>\migration\utilities\sklmv30`

   Default location is `C:\Program Files\IBM\SKLMV301\migration\utilities\sklmv30`.

   Linux `<SKLM_INSTALL_HOME>/migration/utilities/sklmv30`

   Default location is `/opt/IBM/SKLMV301/migration/utilities/sklmv30`.

2. Edit `restore.properties` in the `sklmv30` 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.

   Windows
   ```
   WAS_HOME=C:\\Program Files\\IBM\\WebSphere\\AppServer
   JAVA_HOME=C:\\Program Files\\IBM\\WebSphere\\AppServer\\java\8.0
   BACKUP_PASSWORD=passw0rd123
   DB_PASSWORD=db2_password
   RESTORE_FILE=<SKLM_DATA>\\sklm_v3.0.0.0_20170429013250-0400_migratio
   WAS_USER_PWD=wasadmin_password
   RESTORE_USER_ROLES=y
   #pkcs11_config=C:\\luna.cfg
   ```

   Linux
   ```
   WAS_HOME=/opt/IBM/WebSphere/AppServer
   JAVA_HOME=/opt/IBM/WebSphere/AppServer/java/8.0
   BACKUP_PASSWORD=passw0rd123
   DB_PASSWORD=db2_password
   RESTORE_FILE=<SKLM_DATA>/sklm_v3.0.0.0_20170429013250-0400_migration
   WAS_USER_PWD=wasadmin_password
   RESTORE_USER_ROLES=y
   #pkcs11_config=/luna.cfg
   ```
   Note:

   - To log on to IBM Security Key Lifecycle Manager by using the user credentials that are specified during product installation, set the **RESTORE_USER_ROLES** property set as “n ”. Setting the property to “n ” ensures that user ID and the password are not overwritten with the user credentials of the older version.

     `RESTORE_USER_ROLES=n`
   - If IBM Security Key Lifecycle Manager is configured with HSM, uncomment the **#pkcs11_config** property and specify the correct path of `luna.cfg` file as the value.
   - On Windows operating system, when you specify path in the properties file, use either “/ ” or “\\ ” as path separator as shown in the following example.

     `C:\\sklmv30_restore`

     Or

     `C:/sklmv30_restore`

3. Open a command prompt and run the restore utility.

   Windows
   Go to the `<SKLM_INSTALL_HOME>\migration\utilities\sklmv30` directory and run the following command:

   `restoreV30.bat`

   Linux Go to the `<SKLM_INSTALL_HOME>/migration/utilities/sklmv30` directory and run the following command:

   `restoreV30.sh`

4. Restart IBM Security Key Lifecycle Manager server.

Note: By using the migration restore script, you can restore users, groups, and roles from IBM Security Key Lifecycle Manager, Version

What to do next

Note: After data restoration, ensure that the path for the properties in the `SKLMConfig.properties`, `datastore.properties`, and `ReplicationSKLMConfig.properties` files are correct before you proceed with your next task.

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."

Restoring rollover certificates and key groups

Rollovers that are configured for `LTO` key groups and `3592` certificates are not automatically restored from the earlier versions. To specify these rollovers manually in version 3.0.1, 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\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.

   Note: On Windows operating systems, you must add double slash in the path and specify the path within double quotation marks if the path contains spaces, for example,

   ```
   recreatetask.bat "C:\\Program Files\\IBM\\WebSphere\\AppServer" SKLMAdmin SKLMAdminPwd
   "C:\\Program Files\\IBM\\WebSphere\\AppServer\\products\\sklm\\config\\scheduledTasks.txt"
   "C:\\Program Files\\IBM\\WebSphere\\AppServer\\products\\sklm\\logs\\rolloverlogs.txt" "C:\\Progr
   ```

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 can use the automated backup script to back up files.

Before you begin

You must follow these guidelines before you restore HSM-based encryption backups:

- Ensure that the same HSM partition is present with all its key entries intact on the system where the backup file is restored.
- Master key that was used for the backup key encryption must be intact to restore the backup file. If the master key is refreshed, all the older backups are inaccessible or unusable.
- You must connect to the same HSM and the master key for backup and restore operations irrespective of whether you use HSM-based encryption or password-based encryption.

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\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 **`klmBackup`** 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 **`klmBackup`** 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.

Note: If HSM-based encryption is used for the backups, you need not specify the password.

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 **`tklm.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_DATA`*`/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 parameter. 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 **`tklm.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 **`tklm.backup.db2.dir`** property does not exist.

4. Run the script:
   - Immediately. Type:

     Windows

     ```
     drive:\Program Files\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.

Replication configuration

You can use IBM Security Key Lifecycle Manager to automatically replicate your key materials, configuration files, and other critical information from a primary master server up to 20 secondary clone servers. The automatic replication ensures continuous key and certificate availability to encrypting devices.

The data 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.

The automatic replication ensures the availability of a backup system when the primary IBM Security Key Lifecycle Manager instance is not available. The backup system contains all the required keys and associated data. You can use graphical user interface, CLI commands, or REST interfaces to set up the IBM Security Key Lifecycle Manager automated clone replication process.

Master server configuration

Master server is the primary system that is being replicated. Replication process is triggered only when the new keys, and devices are added or modified on 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.

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

Encryption methods to back up data for replication activities

IBM Security Key Lifecycle Manager supports the following encryption methods for backups:

Password-based encryption

When you configure the master server for automated replication, a password is specified to encrypt the backup key. When data is replicated on the clone server, the same encryption password is used to decrypt and restore the backup files.

HSM-based encryption

You can configure IBM Security Key Lifecycle Manager to use Hardware Security Module (HSM) for storing the master encryption key on master and clone servers. When you run the replication program, the backup key on the master server is encrypted by the master key, which is stored in HSM. When data is replicated on the clone server, the master key in HSM decrypts the backup key. Backup key is used to restore the backup contents.

For more information about encryption methods for backup and replication, see Backup encryption methods for replication activities.

Backup and replication of large amount of data

You can configure IBM Security Key Lifecycle Manager for high-performance backup and replication activities by setting the following parameter in the `SKLMConfig.properties` configuration file of the master server.

```
enableHighScaleBackup=true
```

Note: If you set the **`enableHighScaleBackup=true`** parameter for backup and replication of large amount keys, the master and clone servers must be identical for data replication to be successful. The operating system, directory structures, and DB2 admin user must be same on master and clone servers.

Replication configuration files

You can run IBM Security Key Lifecycle Manager replication as a stand-alone task. A valid replication configuration file must be available to start the automated replication process when the new keys are added.

IBM Security Key Lifecycle Manager uses properties in the `<`*`SKLM_HOME`*`>\config\ReplicationSKLMConfig.properties` configuration file to control the replication process. For example,

Windows
`C:\Program Files\IBM\WebSphere\AppServer\products\sklm\config\ReplicationSKLMConfig.properties`

Linux `/opt/IBM/WebSphere/AppServer/products/sklm/config/ReplicationSKLMConfig.properties`

You can use the IBM Security Key Lifecycle Manager graphical user interface, command-line interface, or REST interface to change properties of the replication configuration file.

You can classify each system as:
- Master - the primary system that is being replicated.
- Clone - the secondary system that is being copied to.

The replication file of the master system can specify up to 20 clones. Each clone system is identified through an IP address or host name, and a port number. You can replicate IBM Security Key Lifecycle Manager environments to multiple clone servers in a manner that is independent of operating systems and directory structure of the server.

Notes:
- Scheduled replication takes place only when the new keys, and devices are added or modified on the master system.
- There can be only one master system with a maximum of 20 clones. Multiple masters are not supported.

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.

Master configuration file sample

```
replication.role=master
replication.auditLogName=replication.log
replication.MaxLogFileSize=1000
replication.MaxBackupNum=10
replication.MaxLogFileNum=3
replication.BackupDestDir=C:\\IBM\WebSphere\\AppServer\\products\\sklm\\restore
backup.ClientIP1=myhost1
backup.ClientPort1=2222
backup.EncryptionPassword=password
backup.ReleaseKeysOnSuccessfulBackup=false
backup.CheckFrequency=24
backup.TLSCertAlias=ssl_cert
replication.MasterListenPort=1111
```

- *master* is the default replication role. Specify role by using the **replication.role** parameter.
- Specify at least one clone with the **backup.ClientIPn** and **backup.ClientPortn** parameter to replicate data to the clone server. For automatically backing up master server data at regular intervals, you need not specify the clone IP address and port.
- Ensure that the specified ports are available and are not currently in use by IBM Security Key Lifecycle Manager or by any other processes.
- You can specify a maximum of 20 clone systems.
- The **backup.TLSCertAlias** parameter must specify a certificate that exists on the master and all clone systems.
- Specify a password to encrypt and decrypt backups. This password becomes obfuscated in the replication configuration file after IBM Security Key Lifecycle Manager reads it for the first time.

Clone configuration file sample

```
replication.role=clone
replication.MasterListenPort=1111
replication.BackupDestDir=C:\\IBM\WebSphere\\AppServer\\products\\sklm\\restore
replication.MaxLogFileSize=1000
replication.MaxBackupNum=3
replication.MaxLogFileNum=4
restore.ListenPort=2222
```

- On the clone system, specify the parameter value `replication.role=clone`.
- The **restore.ListenPort** parameter must specify the port number that is specified in the **backup.ClientIPn** parameter on the master system.

For complete details of all the available replication configuration parameters, see Replication configuration parameters.

Inter-server communication

The Transport Layer Security (TLS) protocol is used for secure communication between the master and clone systems.

An existing private key must be available in the IBM Security Key Lifecycle Manager keystore of the master and all its clone systems. You must set alias of this key on the master system in the **backup.TLSCertAlias** parameter of `ReplicationSKLMConfig.properties` configuration file. If the same key is not available on both the master and clone systems, you cannot start communication between the systems to run the replication task. You can use the graphical user interface, command-line interface, or REST interface to change properties of the replication configuration file.

Replication schedules

Configure the `ReplicationSKLMConfig.properties` file properties for scheduling the automated replication process.

Use the graphical user interface, command-line interface, or REST interface to configure properties of the replication configuration file for scheduling replication process. Scheduled replication takes place only when the new keys, and devices are added or modified on the master server. 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.

You can configure the schedule so that IBM Security Key Lifecycle Manager checks whether the replication is required periodically, and starts the process if changes are made. You can also specify a time of day to run a replication when required. Configure the **`backup.CheckFrequency`** parameter to specify how often IBM Security Key Lifecycle Manager checks the master system for updates. Replication triggers when the updates take place. The value is set in minutes with 1440 as the default value.

To specify a time of day, configure the **`backup.DailyStartReplicationBackupTime`** parameter. You must specify a time in 24-hour clock format (HH:MM). Replication takes place only when the master system changes since the last replication.

By default, the clone system restores a backup as soon it is received from the master system. To specify the restoration time, add the **`restore.DailyStartReplicationRestoreTime`** parameter in the replication configuration file of the clone system. You must specify time in 24-hour clock format (HH:MM).

You can use the **`tklmReplicationNow`** CLI command or **`Replication Now REST Service`** to force an ad hoc replication to all the defined clones, or a specific replication.

Replication audit records

IBM Security Key Lifecycle Manager replication records audit information to the IBM Security Key Lifecycle Manager audit log file.

IBM Security Key Lifecycle Manager replication program provides a facility to write replication-specific audit records to its own discrete audit log file. Replication audit log records all the actions that are related to replication process. By default, location of the replication audit log file is `<`*`SKLM_HOME`*`>\logs\replication\replication_audit.log`.

Use the graphical user interface, command-line interface, or REST interface to set audit properties in the `ReplicationSKLMConfig.properties` file. In the configuration file, you can configure audit properties, such as audit log file location, log file name, log file size, maximum number of log files to keep, or maximum number of backup files to keep.

Configuring a master server with password-based encryption for backups

You can change default settings of master server for communication with the clone server to replicate IBM Security Key Lifecycle Manager data by using password-based encryption for the backups.

About this task

Note: Data is replicated to the clone servers based on a configured schedule only when new cryptographic objects are added to the master server.

Use the graphical user interface, command-line interface, or REST interface to change the settings in the `ReplicationSKLMConfig.properties` configuration file according to your requirements.

For more information about encryption methods for backups and replication, see Backup encryption methods for replication activities.

If you want to configure IBM Security Key Lifecycle Manager for high performance backup and replication activities, you must set following parameter in the <*SKLM_HOME*>`/config/SKLMConfig.properties` configuration file of the master server, for example, `C:\Program Files\IBM\WebSphere\AppServer\products\sklm\config\SKLMConfig.properties`.

```
enableHighScaleBackup=true
```

Note: If you set the **`enableHighScaleBackup=true`** parameter for backup and replication of large amount keys, the master and clone systems must be identical for data replication to be successful. The operating system, directory structures, and DB2 admin user must be same on master and clone servers.

Procedure

1. Go to the appropriate page or directory:

   Graphical user interface

   a. Log on to the graphical user interface.

   b. Click **IBM Security Key Lifecycle Manager** > **Administration** > **Replication**.

   Command-line interface

   a. Go to the `WAS_HOME/bin` directory. For example,

      Windows

      ```
      cd drive:\Program Files (x86)\IBM\WebSphere\AppServer\bin
      ```

      Linux

      ```
      cd /opt/IBM/WebSphere/AppServer/bin
      ```

   b. Start the **`wsadmin`** interface by using an authorized user ID, such as `SKLMAdmin`. For example,

      Windows

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

      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:

   Graphical user interface

   a. Select **Master**.
   b. Select a replication server management option.

      Start Replication Server
      Click **Start Replication Server** to start the replication server for replicating the current IBM Security Key Lifecycle Manager active files and data on clone servers based on a configured schedule.

      Stop Replication Server
      Click **Stop Replication Server** to stop the replication server so that the current IBM Security Key Lifecycle Manager active files and data are not replicated to the clone servers.

      Replicate Now
      Click **Replicate Now** to immediately run the IBM Security Key Lifecycle Manager replication task, and to force a backup to be sent to the configured clones.
   c. Configure the settings.

      Basic Properties

Certificate from keystore	Select a certificate from the list. Ensure that SSL/TLS certificate exists 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. Clone server uses the same password to decrypt and restore the file. **Note:** If HSM-based encryption is used for the backups, you need not specify the password.
Confirm replication backup encryption passphrase	Specify the same password again to verify the password that you specified.
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. The **Replication backup destination directory** field displays the default <*SKLM_DATA*> directory path, where the backup file is saved for example, `C:\Program Files\IBM\WebSphere\AppServer\products\sklm\data`. For the definition of <*SKLM_DATA*>, see Definitions for *HOME* and other directory variables. Click **Browse** to specify a backup repository location under <*SKLM_DATA*> directory. Directory path in the **Replication backup destination directory** field changes based on the value that is set for the **`browse.root.dir`** property in the `SKLMConfig.properties` file.
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 24 hours. 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.

d. 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 run **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:<port>/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:<port>/SKLM/rest/v1/configProperties
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth authId=139aeh34567m
Accept-Language: en
{ "replication.role": "master"}
```

What to do next

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

Configuring a master server with password-based encryption when HSM is configured

You can change default settings of master server for communication with the clone server to replicate IBM Security Key Lifecycle Manager data by using password-based encryption for the backups when Hardware Security Module (HSM) is configured.

Before you begin

Ensure that IBM Security Key Lifecycle Manager is configured to use HSM for storing the master key before you back up and replicate data with HSM-based encryption. For the configuration steps, see “Configuring HSM parameters” on page 194.

About this task

Note: Data is replicated to the clone servers based on a configured schedule only when new cryptographic objects are added to the master server.

Use the graphical user interface, command-line interface, or REST interface to change the settings in the ReplicationSKLMConfig.properties configuration file according to your requirements.

You must set the **enablePBEInHSM=true** property in the *<SKLM_HOME>*/config/SKLMConfig.properties file to back up data with password-based encryption when HSM is configured.

For more information about encryption methods for backups and replication, see Backup encryption methods for replication activities.

If you want to configure IBM Security Key Lifecycle Manager for high performance backup and replication activities, you must set the **enableHighScaleBackup=true** parameter in the *<SKLM_HOME>*/config/SKLMConfig.properties file configuration file of the master server.

Note: If you set the **enableHighScaleBackup=true** parameter for backup and replication of large amount keys, the master and clone systems must be identical for data replication to be successful. The operating system, directory structures, and DB2 admin user must be same on master and clone servers.

Procedure

1. Set the **enablePBEInHSM=true** property in the *<SKLM_HOME>*/config/SKLMConfig.properties file.

   Command-line interface

   a. Go to the WAS_HOME/bin directory. For example,

      Windows

      ```
      cd drive:\Program Files\IBM\WebSphere\AppServer\bin
      ```

      Linux

      ```
      cd /opt/IBM/WebSphere/AppServer/bin
      ```

   b. Start the **wsadmin** interface by using an authorized user ID, such as SKLMAdmin. For example,

      Windows

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

      Linux

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

   c. Run the **tklmConfigUpdateEntry** command to set **enablePBEInHSM** property in the SKLMConfig.properties configuration file.

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

   REST interface

   a. Open a REST client.

   b. 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.

   c. Run **Update Config Property REST Service** to set **enablePBEInHSM** property in the SKLMConfig.properties configuration file. Pass the user authentication identifier that you obtained in Step b along with the request message as shown in the following example.

      ```
      PUT https://localhost:<port>/SKLM/rest/v1/configProperties
      Content-Type: application/json
      Accept : application/json
      Authorization: SKLMAuth userAuthId=139aeh34567m
      Accept-Language : en
      { "enablePBEInHSM" : "true"}
      ```

2. Go to the appropriate page or directory to configure replication parameters.

   Graphical user interface

   a. Log on to the graphical user interface.

b. Click **IBM Security Key Lifecycle Manager** > **Administration** > **Replication**.

Command-line interface

a. Go to the `WAS_HOME/bin` directory.

b. Start the **`wsadmin`** interface by using an authorized user ID, such as `SKLMAdmin`.

REST interface

Open a REST client.

3. Change the value for one or more settings of the master server.

Graphical user interface

a. Select **Master**.

b. Select a replication server management option.

Start Replication Server

Click **Start Replication Server** to start the replication server for replicating the current IBM Security Key Lifecycle Manager active files and data on clone servers based on a configured schedule.

Stop Replication Server

Click **Stop Replication Server** to stop the replication server so that the current IBM Security Key Lifecycle Manager active files and data are not replicated to the clone servers.

Replicate Now

Click **Replicate Now** to immediately run the IBM Security Key Lifecycle Manager replication task, and to force a backup to be sent to the configured clones.

c. Specify the appropriate settings.

Basic Properties

Certificate from keystore	Select a certificate from the list. Ensure that SSL/TLS certificate exists 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. Clone server uses the same password to decrypt and restore the file.
Confirm replication backup encryption passphrase	Specify the same password again to verify the password that you specified.
Master listen port	Port number for communication when unserialized or delayed replications take place. Default master listen port is 1111.
Click the **Add Clone** link in the **Clone Details** section to configure replication settings for clones.	
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.
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. The **Replication backup destination directory** field displays the default <*SKLM_DATA*> directory path, where the backup file is saved, for example, `C:\Program Files\IBM\WebSphere\AppServer\products\sklm\data`. For the definition of <*SKLM_DATA*>, see Definitions for *HOME* and other directory variables. Click **Browse** to specify a backup repository location under <*SKLM_DATA*> directory. Directory path in the **Replication backup destination directory** field changes based on the value that is set for the **`browse.root.dir`** property in the `SKLMConfig.properties` file.
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 24 hours. 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.

d. 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 as shown in the following example.

```
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]')
```

For complete details of all the available replication configuration parameters, see Replication configuration parameters.

REST interface

a. Run **`Get Single Config Property REST Service`** by sending the HTTP GET request as shown in the following example.

Service request

```
GET https://localhost:<port>/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"}
```

b. 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:<port>/SKLM/rest/v1/configProperties
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth authId=139aeh34567m
Accept-Language: en
{ "replication.role": "master"}
```

For complete details of all the available replication configuration parameters, see Replication configuration parameters.

What to do next

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

Configuring a master server with HSM-based encryption for backups

You can change default settings of master server for communication with the clone server to replicate IBM Security Key Lifecycle Manager data by using HSM-based encryption for the backups.

Before you begin

Ensure that IBM Security Key Lifecycle Manager is configured to use HSM for storing the master key before you back up and replicate data with HSM-based encryption. For the configuration steps, see "Configuring HSM parameters" on page 194.

Consider the following guidelines for using HSM-based encryption.

- Same HSM partition must be present with all its key entries intact on all the clone servers.
- Master key that you used for the backup key encryption must be intact to replicate the backup file on the clone server. If the master key is refreshed, all the older backups are inaccessible or unusable.
- You must connect to the same HSM and the master key for automated replication irrespective of whether you use HSM-based encryption or password-based encryption.

About this task

Note: Data is replicated to the clone servers based on a configured schedule only when new cryptographic objects are added to the master server.

Use the graphical user interface, command-line interface, or REST interface to change the settings in the `ReplicationSKLMConfig.properties` configuration file according to your requirements.

For more information about encryption methods for backups and replication, see Backup encryption methods for replication activities.

If you want to configure IBM Security Key Lifecycle Manager for high performance backup and replication activities, you must set following parameter in the `<SKLM_HOME>\config\SKLMConfig.properties` configuration file of the master server, for example, `C:\Program Files\IBM\WebSphere\AppServer\products\sklm\config\SKLMConfig.properties`.

```
enableHighScaleBackup=true
```

Note: If you set the **`enableHighScaleBackup=true`** parameter for backup and replication of large amount keys, the master and clone systems must be identical for data replication to be successful. The operating system, directory structures, and DB2 admin user must be same on master and clone servers.

Procedure

1. Go to the appropriate page or directory:

   Graphical user interface

   a. Log on to the graphical user interface.
   b. Click **IBM Security Key Lifecycle Manager** > **Administration** > **Replication**.

   Command-line interface

   a. Go to the `WAS_HOME/bin` directory. For example,

      Windows
      ```
      cd drive:\Program Files (x86)\IBM\WebSphere\AppServer\bin
      ```
      Linux
      ```
      cd /opt/IBM/WebSphere/AppServer/bin
      ```
   b. Start the **`wsadmin`** interface by using an authorized user ID, such as `SKLMAdmin`. For example,

      Windows
      ```
      wsadmin.bat -username SKLMAdmin -password mypwd -lang jython
      ```
      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:

   Graphical user interface

   a. Select **Master**.
   b. Select a replication server management option.

      Start Replication Server

      Click **Start Replication Server** to start the replication server for replicating the current IBM Security Key Lifecycle Manager active files and data on clone servers based on a configured schedule.

Stop Replication Server
Click **Stop Replication Server** to stop the replication server so that the current IBM Security Key Lifecycle Manager active files and data are not replicated to the clone servers.

Replicate Now
Click **Replicate Now** to immediately run the IBM Security Key Lifecycle Manager replication task, and to force a backup to be sent to the configured clones.

c. Specify the appropriate settings:

Basic Properties

Certificate from keystore	Select a certificate from the list. Ensure that SSL/TLS certificate exists on the master and all clone systems that you configure for replication.
Master listen port	Port number for communication when unserialized or delayed replications take place. Default master listen port is `1111`.
Click the **Add Clone** link in the **Clone Details** section to configure replication settings for clones.	
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. The **Replication backup destination directory** field displays the default <*SKLM_DATA*> directory path, where the backup file is saved, for example, `C:\Program Files\IBM\WebSphere\AppServer\products\sklm\data`. For the definition of <*SKLM_DATA*>, see Definitions for *HOME* and other directory variables. Click **Browse** to specify a backup repository location under <*SKLM_DATA*> directory. Directory path in the **Replication backup destination directory** field changes based on the value that is set for the **`browse.root.dir`** property in the `SKLMConfig.properties` file.
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 24 hours. 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.

d. 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]')
```

For complete details of all the available replication configuration parameters, see Replication configuration parameters.

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 run **`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:<port>/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:<port>/SKLM/rest/v1/configProperties
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth authId=139aeh34567m
Accept-Language: en
{ "replication.role": "master"}
```

For complete details of all the available replication configuration parameters, see Replication configuration parameters.

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 a clone server to receive IBM Security Key Lifecycle Manager data from master server based on a configured schedule. Data is replicated to the clone servers only when new cryptographic objects are added to the master 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** > **Administration** > **Replication**.
   - Command-line interface
     a. Go to the `<`*`WAS_HOME`*`>/bin` directory. For example,

        Windows
        ```
        cd drive:\Program Files\IBM\WebSphere\AppServer\bin
        ```
        Linux
        ```
        cd /opt/IBM/WebSphere/AppServer/bin
        ```
     b. Start the **`wsadmin`** interface by using an authorized user ID, such as `SKLMAdmin`. For example,

        Windows
        ```
        wsadmin.bat -username SKLMAdmin -password mypwd -lang jython
        ```
        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. Select a replication server management option.

Start Replication Server
Click **Start Replication Server** to start the replication server for receiving IBM Security Key Lifecycle Manager data from the master server.

Stop Replication Server
Click **Stop Replication Server** to stop the replication server so that IBM Security Key Lifecycle Manager data from the master server is not received by the clone servers.

c. 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 `<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.

d. 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:<port>/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:<port>/SKLM/rest/v1/configProperties
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth authId=139aeh34567m
Accept-Language: en
{ "replication.role": "clone"}
```

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 graphical user interface, command-line interface, or REST interface to change the settings in the ReplicationSKLMConfig.properties configuration file according to your requirements.

IBM Security Key Lifecycle Manager supports the following encryption methods for backups:

Password-based encryption
When you configure the master server for automated replication, a password is specified to encrypt the backup key. When data is replicated on the clone server, the same encryption password is used to decrypt and restore the backup files.

HSM-based encryption
You can configure IBM Security Key Lifecycle Manager to use Hardware

Security Module (HSM) for storing the master encryption key on master and clone servers. When you run the replication program, the backup key on the master server is encrypted by the master key, which is stored in HSM. When data is replicated on the clone server, the master key in HSM decrypts the backup key. Backup key is used to restore the backup contents.

For information about how to configure the master server with HSM-based encryption, see "Configuring a master server with HSM-based encryption for backups" on page 175. For information about how to configure the master server with password-based encryption when HSM is configured, see "Configuring a master server with password-based encryption when HSM is configured" on page 171.

The following procedure describes how to schedule automatic backup operation by using password-based encryption.

Procedure

1. Go to the appropriate page or directory:

   Graphical user interface

   a. Log on to the graphical user interface.
   b. Click **IBM Security Key Lifecycle Manager** > **Administration** > **Replication**.

   Command-line interface

   a. Go to the `WAS_HOME/bin` directory. For example,

      Windows

      ```
      cd drive:\Program Files\IBM\WebSphere\AppServer\bin
      ```

      Linux

      ```
      cd /opt/IBM/WebSphere/AppServer/bin
      ```

   b. Start the **`wsadmin`** interface by using an authorized user ID, such as `SKLMAdmin`. For example,

      Windows

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

      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:

   Graphical user interface

   a. Select **Master**.
   b. Select a replication server management option.

      Start Replication Server

      Click **Start Replication Server** to start the replication server for backing up IBM Security Key Lifecycle Manager data based on a configured schedule.

      Stop Replication Server

      Click **Stop Replication Server** to stop the replication server so that the IBM Security Key Lifecycle Manager data is not backed up.

Replicate Now
Click **Replicate Now** to immediately run the IBM Security Key Lifecycle Manager replication task, and to force a backup file creation.

c. Configure the settings.

Basic Properties

Certificate from keystore	Select a certificate from the list. Ensure that SSL/TLS certificate exists 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. **Note:** If HSM-based encryption is used for the backups, you need not specify the password.
Confirm replication backup encryption passphrase	Specify the same password again to verify the password that you specified.
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. The **Replication backup destination directory** field displays the default *<SKLM_DATA>* directory path, where the backup file is saved, for example, `C:\Program Files\IBM\WebSphere\AppServer\products\sklm\data`. For the definition of *<SKLM_DATA>*, see Definitions for *HOME* and other directory variables. Click **Browse** to specify a backup repository location under *<SKLM_DATA>* directory.
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.

d. 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 run **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:<port>/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:<port>/SKLM/rest/v1/configProperties
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth authId=139aeh34567m
Accept-Language: en
{ "replication.role": "master"}
```

Setting up replication process by using CLI commands and REST services

You must set up a basic environment in IBM Security Key Lifecycle Manager to run the replication process.

About this task

This topic describes how to set up replication process by using the IBM Security Key Lifecycle Manager command-line interface commands and the REST interfaces for replication.

Procedure

1. Set up the IBM Security Key Lifecycle Manager master system.

2. Specify a SSLSERVER certificate for the replication to work. You can create the certificate by using graphical user interface, command-line interface, or the REST interface as shown in the following examples.

   Graphical user interface

   a. Log on to the graphical user interface.
   b. Click **Advanced Configuration** > **Server Certificates**.

   Command-line interface

   a. Go to the WAS_HOME/bin directory. For example,

      Windows

      ```
      cd drive:\Program Files\IBM\WebSphere\AppServer\bin
      ```

      Linux

      ```
      cd /opt/IBM/WebSphere/AppServer/bin
      ```

   b. Start the **wsadmin** interface by using an authorized user ID, such as SKLMAdmin. For example,

      Windows

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

      Linux

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

   c. 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]')
      ```

   REST interface

   a. Open a REST client.
   b. 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.
   c. To create a self-signed certificate, run **Certificate Generate Request REST Service** by sending the following HTTP request.

      ```
      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 "  }
      ```

3. Create a backup of the master IBM Security Key Lifecycle Manager as shown in the following examples.

   Note: You need not specify the password when IBM Security Key Lifecycle Manager is configured to use Hardware Security Module (HSM) for storing the master encryption key. For information about encryption methods to back up data, see HSM-based encryption for backups.

   Graphical user interface

   a. Log on to the graphical user interface.
   b. Click **Backup and Restore**.

Command-line interface

Type the **tklmBackupRun** command.

```
print AdminTask.tklmBackupRun
('[-backupDirectory C:\\wasbak1\\sklmbackup1 -password myBackupPwd]')
```

REST interface

To create a backup, run **Backup Run REST Service** by sending the following HTTP request.

```
POST https://localhost:<port>/SKLM/rest/v1/ckms/backups
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth authId=139aeh34567m
Accept-Language : en
{"backupDirectory":"/sklmbackup1","password":"myBackupPwd"}
```

Note:

You can configure IBM Security Key Lifecycle Manager for high performance backup and replication activities by setting the following parameter in the `SKLMConfig.properties` configuration file of the master server.

```
enableHighScaleBackup=true
```

To back up and replication of large amount keys, the master and clone servers must be identical. The operating system, directory structures, and DB2 admin user must be same on the servers.

For information about how to back up large amount of data, see “Backing up large amount of data” on page 121.

4. Take the backup that is created in Step 3 and copy it to each of your IBM Security Key Lifecycle Manager clone systems. Restore this backup to each of these systems as shown in the following examples:

Graphical user interface

a. Log on to the graphical user interface.

b. Click **Backup and Restore**.

Command-line interface

Type the **tklmBackupRestoreRun** command on one line:

```
print AdminTask.tklmBackupRunRestore
('[-backupFilePath /opt/sklmbackup/sklm_v2.7_20160412074433_backup.jar
  -password myBackupPwd]')
```

REST interface

To restore a backup, run **Backup Run Restore REST Service** by sending the following HTTP request.

```
POST https://localhost:<port>/SKLM/rest/v1/ckms/restore
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth authId=139aeh34567m
Accept-Language : en
{"backupFilePath":"/sklmbackup","password":"myBackupPwd"}
```

5. Create the `ReplicationSKLMConfig.properties` replication configuration file on the master system. This configuration file must be a text file and you must locate the file in the same directory as the IBM Security Key Lifecycle Manager properties file, for example `C:\Program Files\IBM\WebSphere\AppServer\products\sklm\config\ReplicationSKLMConfig.properties`.

   Use command-line interface or REST services to set properties in the `ReplicationSKLMConfig.properties`.

Command-line interface

Type the **tklmReplicationConfigUpdateEntry** command on one line to set the value of the **replication.role** property to master.

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

REST interface

To set the value, run **Update Replication Config Property REST Service** by sending the following HTTP request.

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

For complete details of all the available replication configuration parameters, see Replication configuration parameters.

The following example shows the fields that are mandatory on the master to allow the replication task to start.

- Set role to master.
- Identify certificate from Step 2 and provide at least one clone server and port number.
- Define a master listen port and choose a password.

  Note: You need not specify the password when IBM Security Key Lifecycle Manager is configured to use Hardware Security Module (HSM) for storing the master encryption key. For information about encryption methods to back up data for replication activities, see Backup encryption methods for replication activities.

```
backup.EncryptionPassword=mypassword
backup.TLSCertAlias=sklmSSLCertificate
backup.ClientIP1=myhostname
backup.ClientPort1=2222
replication.MasterListenPort=1111
```

The **backup.EncryptionPassword** property can contain of characters, numbers, or special characters. The product obfuscates this property when replication is first run. The **backup.TLSCertAlias** property specifies the alias of the certificate and the private key that is used to communicate to the clone created in Step 2.

The **replication.MasterListenPort** property specifies the port that the master system listens on for certain responses from the clones. The **backup.ClientIP1** and the **backup.ClientPort1** properties define the clone. The **backup.ClientIP1** property can be either a host name or an IP address. The **backup.ClientPort1** property specifies the port that the client is listening on. To define other clones, you must specify the **backup.ClientIP*** and **backup.ClientPort*** properties, where "*" is a number 2 - 20, like you did for the first set.

6. Create the ReplicationSKLMConfig.properties replication configuration file on the clone system. This configuration file must be a text file and you must locate the file in the same directory as the IBM Security Key Lifecycle Manager properties file, for example, C:\Program Files\IBM\WebSphere\AppServer\products\sklm\config\ReplicationSKLMConfig.properties.

   Use the command-line interface or REST interface to set the properties in the replication configuration file as described in Step 5.

   The following example shows the fields that are mandatory on the clone to allow the replication task to start.

- Set role to clone.
- Define a master listen port.
- Define a restore listen port. The port must be the same port number that is coded in the corresponding **backup.ClientPort*** parameter on the master server.

```
replication.role=clone
backup.TLSCertAlias=sklmSSLCertificate
replication.MasterListenPort=1111
restore.ListenPort=2222
```

Setting the **replication.role** property is mandatory for clones. By default, the value of this property is master. The **backup.TLSCertAlias** property must set to the certificate created in Step 2. This property is used to send the status of the clone when replication is postponed for a later time, or the restore process takes longer than the master is waiting for a response.

The **replication.MasterListenPort** property specifies the port to send status when replication is postponed for a later time, or the restore process takes longer than the master is waiting for a response. The last property **restore.ListenPort** is the port that the clone listens on for replication requests from the master.

7. You can run ad-hoc replication task by using the **TKLMReplicationNow** CLI command or **Replication Now REST Service**. Or, you can setup a scheduled replication. You can schedule a backup task by using the **backup.DailyStartReplicationBackupTime** or **backup.CheckFrequency** property in the replication configuration file.

   The following property schedules the backup task at 11 PM everyday.

   ```
   backup.DailyStartReplicationBackupTime=23:00
   ```

   The following property checks whether a backup is required for every 24 hours and runs the task if required.

   ```
   backup.CheckFrequency=1440
   ```

8. Restart IBM Security Key Lifecycle Manager on master and clone systems. You can see the following messages on a clone and master system: Use the **tklmReplicationStatus** CLI command to ensure that the replication task is running. You can see the following messages on a master and a clone system:

   Command-line interface

   You can use the following CLI command to ensure that the replication task is running:

   ```
   print AdminTask.tklmReplicationStatus()
   ```

   Master system

   ```
   CTGKM2215I The Security Key Lifecycle Manager Replication
   task is UP. Role set to: MASTER
   CTGKM2218I The last completed replication took place at
   Thu Jun 19 14:50:59 WST 2017
   CTGKM2217I The next scheduled replication is due at
   Fri Jun 20 17:03:36 WST 2017
   ```

   Clone system

   ```
   CTGKM2215I The SKLM Replication task is UP. Role set to: CLONE
   CTGKM2220I No previous successful replications.
   CTGKM2221I No replication currently scheduled.
   ```

   REST interface

   Use **Replication Status REST Service** to ensure that the replication task is running. Send the following HTTP request by using a REST client:

```
GET https://localhost:<port>/SKLM/rest/v1/replicate/status
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth authId=139aeh34567m
```

Master system

```
Status Code : 200 OK
Content-Language: en
[
  {code:"CTGKM2215I", "status":"CTGKM2215I The Security Key
Lifecycle Manager  Replication task is UP. Role set to: MASTER"},
  {code:"CTGKM2218I", "status":"CTGKM2218I The last completed
replication took place at Thu Jun 19 14:50:59 WST 2015."} ,
  {code:"CTGKM2217I", "status":"CTGKM2217I The next scheduled
replication is due at Fri Jun 20 17:03:36 WST 2015." }

]
```

Clone system

```
Status Code : 200 OK
[
  { code:"CTGKM2215I", "status":"CTGKM2215I The Security Key
Lifecycle Manager  Replication task is UP. Role set to: CLONE"}
, { code:"CTGKM2220I", "status":"CTGKM2220I No previous
   successful replications."} ,
  { code:"CTGKM2217I", "status":"CTGKM2221I No replication
   currently scheduled." }

]
```

9. Replication is now set up and replication checks for changes every 60 minutes. You can change this interval, set up a certain time every day for replication to check for changes. You can also use the **tklmReplicationNow** CLI command or **Replication Now REST Service** to run a replication task immediately.

Replication problems and resolution

You must consider possible issues on the clone and master systems when you run the IBM Security Key Lifecycle Manager replication task.

Incomplete replication

- Ensure that the TSL/SSL certificate with private key that is specified in the **backup.TLSCertAlias** parameter are available on both the master and clone servers.
- Ensure that port number for the master server is free. Clone port numbers that are configured on the master server must be free on the clone server.
- Check the server names or IP addresses specified in the replication configuration file are correct and accessible from the master server.
- Check whether the replication task is up on each server by running the **tklmReplicationStatus** command, **Replication Status REST Service**, or the status on the **Replication** section of IBM Security Key Lifecycle Manager welcome page.
- For DB2 replication, ensure that date/time of master and clone servers are closely synchronized. Large discrepancies can lead to restore failure.
- Check the replication configuration file to ensure that the minimum required parameters are defined, without typographical error.
- Define a maximum of 1 master and 20 associated clones.
- Check the replication audit file to get more information about replication failure.

Replication is not taking place at scheduled time

- Scheduled replications take place only when new keys, and devices are added or modified on the master server.
- When both specific replication time and a check interval are set in the master replication configuration file, the time overrides the check interval.

Clone system replication

- The clone IBM Security Key Lifecycle Manager server restarts after replication.
- Maintain the availability of your clone servers. You can specify a specific time-of-day to complete the replication with the **restore.DailyStartReplicationRestoreTime** parameter. For example, to run restores only at 11 PM, regardless of when the backup file is received, code the following property in the configuration file:

  ```
  restore.DailyStartReplicationRestoreTime=23:00
  ```

Restarting the IBM Security Key Lifecycle Manager server

Restart of the server causes the server to read its configuration and accept the configuration changes, if any. To restart the IBM Security Key Lifecycle Manager server, you can run the server restart scripts, REST service, or use the graphical user interface.

About this task

To restart server, use the <*IBM Security Key Lifecycle Manager User*> link on welcome page header bar, **Restart Server REST Service**, or run the **stopServer** and **startServer** scripts.

Procedure

1. Go to the appropriate page or directory.

   Graphical user interface

   a. Log on to the graphical user interface.

   b. On the welcome page header bar, click the <*IBM Security Key Lifecycle Manager User*> link. For example, click the **SKLMAdmin** link.

   Server restart scripts

   a. Go to the <*WAS_HOME*>\bin directory.

   Windows
   `C:\Program Files\IBM\WebSphere\AppServer\bin`

   Linux `/opt/IBM/WebSphere/AppServer/bin`

   REST interface
   Open a REST client.

2. Restart the server.

   Graphical user interface

   a. Click **Restart Server**.

   b. Click **OK**.

   Note: The IBM Security Key Lifecycle Manager server is unavailable for a few minutes while the restart operation is in progress.

Server restart scripts

a. Stop the server.

Windows

```
stopServer.bat server1
```

Linux

```
./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:

Windows

```
stopServer.bat server1 -username wasadmin -password mypwd
```

Linux

```
./stopServer.sh server1  -username wasadmin -password mypwd
```

b. Start the server.

Windows

```
startServer.bat server1
```

Linux

```
./startServer.sh server1
```

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 run **Restart Server 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:<port>/SKLM/rest/v1/ckms/servermanagement/restartServer
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
```

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** > **Global security**.
3. 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.

4. Click **Apply**.
5. Click **Save** in the Messages box.
6. 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** > **Global security**.
3. Clear the **Enable administrative security** check box.
4. Click **Apply**.
5. Click **Save** in the Messages box.
6. 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.

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 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`
- `SafeNet Luna SA 6.1`
- `nCipher nShield Connect 1500`
- `IBM 4765 PCIe Cryptographic Coprocessor`

Note:

- You can use `SafeNet Luna SA 4.5`, `SafeNet Luna SA 5.0`, `SafeNet Luna SA 6.1`, 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:
  - Convert 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 and from a PKCS11IMPLKS (PKCS#11) keystore

You can use the following configuration parameters to define HSM:

- **pkcs11.pin**
- **pkcs11.config**
- **useMasterKeyInHSM**

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, 5.0, and 6.1

```
#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.config**, and **useMasterKeyinHSM** configuration parameters to define Hardware Security Module.

Procedure

1. Set up and configure the HSM as per the instructions from HSM manufacturers.
2. Add the **pkcs11.pin**, **pkcs11.config**, and **useMasterKeyinHSM** 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>]')
   ```

   ```
   print AdminTask.tklmConfigUpdateEntry('[-name useMasterKeyinHSM -value
   <true | false>]')
   ```

   REST interface

   ```
   PUT https://localhost:<port>/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:<port>/SKLM/rest/v1/configProperties
   Content-Type: application/json
   Accept: application/json
   Authorization: SKLMAuth authId=139aeh34567m
   Accept-Language: en
   { "pkcs11.config" : "<hsm config file>"}
   ```

   ```
   PUT https://localhost:<port>/SKLM/rest/v1/configProperties
   Content-Type: application/json
   Accept: application/json
   ```

```
Authorization: SKLMAuth authId=139aeh34567m
Accept-Language: en
{ "useMasterKeyinHSM" : "<true | false>"}
```

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\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. IBM Security Key Lifecycle Manager supports 64-bit HSM client.

- 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, `SafeNet Luna SA 5.0` card, and `SafeNet Luna SA 6.1` 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, `SafeNet Luna SA 5.0` card, and `SafeNet Luna SA 6.1` 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.

To integrate LDAP with IBM Security Key Lifecycle Manager, you can use any of the following configuration methods:

- By using WebSphere Integrated Solutions Console. For more information, see Integrating LDAP by using WebSphere Integrated Solutions Console.
- By running the LDAP configuration scripts. For more information, see Running the LDAP configuration scripts.

LDAP integration by using WebSphere Integrated Solutions Console

Before you integrate LDAP with IBM Security Key Lifecycle Manager by using WebSphere Integrated Solutions Console, you must run the backup tasks.

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 (KLMProfile) 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
      KLMProfile -backupFile <path to a file>
      ```

      ```
      C:\Program Files\IBM\WebSphere\AppServer\bin\manageProfiles.bat
      backupProfile -profileName KLMProfile -backupFile
      :\SKLM_WAS_ProfileBackup
      ```

Linux `<WAS_HOME>/bin/manageprofiles.sh -backupProfile -profileName KLMProfile -backupFile <path to a file>`

```
/opt/IBM/WebSphere/AppServer/bin/manageprofiles.sh -backupProfile
profileName KLMProfile -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.ibm.com/support/knowledgecenter/SSEQTP_9.0.0/com.ibm.websphere.base.doc/ae/rxml_manageprofiles.html.

Integrating LDAP by using WebSphere Integrated Solutions Console

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 195

Procedure

1. Add LDAP repository to the federated repository. For the instructions, see "Adding LDAP repository to the federated repository" on page 198.
2. Create the database for LDAP configuration.
   a. Open the DB2 command window.
   b. Run the following command to create the database.
   ```
   db2  create database USERDB31 using codeset UTF-8 territory US
   ```
3. Update the data source from the WebSphere Integrated Solutions Console with `jndi` name `jdbc/wimXADS`. For the instructions, see "Updating a data source from WebSphere Integrated Solutions Console" on page 199.
4. Restart WebSphere Application Server.
5. Copy `db2jcc.jar and db2jcc_license_cu.jar` from the `DB2SKLMV301` folder to the *`WAS_HOME`*`/lib` folder.

   `DB2SKLMV301` path:

   Windows
   `drive``:\Program Files\IBM\DB2SKLMV301\java`

   Linux *`path`*`/IBM/DB2SKLMV301/java`

   Default definition of *WAS_HOME* variable is typically:

   Windows
   `C:\Program Files\IBM\WebSphere\AppServer`

   Linux `/opt/IBM/WebSphere/AppServer`
6. 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 199.
7. 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 201.

8. Restart WebSphere Application Server.
9. 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 201
10. 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 more information, see “Post-LDAP configuration tasks to support LDAP integration” on page 205

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:9093/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.

Updating a data source from WebSphere Integrated Solutions Console:

You must update 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 update the data source from WebSphere Integrated Solutions Console (`https://localhost:9093/ibm/console/logon.jsp`) as a `wasadmin` user.
2. In the navigation bar, click **Resources** > **JDBC** > **Data sources**.
3. Click **WIM Data Source** to edit the database values.
4. Update the database name with `USERDB31` under the **Common and required data source properties** section.
5. Click **OK**.
6. Click **Save** to save the configuration.

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 `<*SKLM_INSTALL_HOME*>\bin` folder.

Note: All the `.py` python scripts are present in the <*SKLM_INSTALL_HOME*>`\bin\LDAPIntegration` directory.
<*SKLM_INSTALL_HOME*> path typically,

Windows
`C:\Program Files\IBM\SKLMV301`

Linux `opt/IBM/SKLMV301`

2. Run the following commands:

   ```
   wsadmin.bat -user wasadmin user -password wasadmin passwd -lang jython -f
   SKLM_INSTALL_HOME\bin\LDAPIntegration\createDBRepos.py WAS_HOME LDAP_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,

   ```
   LDAP_DBNAME = USERDB31
   SKLM_DBUSER = SKLMDB31
   SKLM_DBPORT# = 50050
   ```

   `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_INSTALL_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:9093/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_INSTALL_HOME>\bin\LDAPIntegration\addGroupsToDBRepos.py
   ```
7. Run the following command.

   ```
   wsadmin.bat -user <wasadmin user> -password <wasadmin passwd> -lang jython
   -f <SKLM_INSTALL_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 `<SKLM_INSTALL_HOME>/bin` folder.

Note: All the `.py` python scripts are present in the `<SKLM_INSTALL_HOME>/bin/LDAPIntegration` directory.
`<SKLM_INSTALL_HOME>` path typically,

Windows
`C:\Program Files\IBM\SKLMV301`

Linux `/opt/IBM/SKLMV301`

2. Run the following commands:

```
wsadmin.bat -user <wasadmin user> -password <waasadmin 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_INSTALL_HOME>\bin\LDAPIntegration\addLDAPUserToGroup.py
<user uniqueName> klmGUICLIAccessGroup
```

What to do next

Take IBM Security Key Lifecycle Manager application backup.

Running the LDAP configuration scripts

Run the LDAP configuration scripts to easily integrate IBM Security Key Lifecycle Manager with LDAP for configuring IBM Security Key Lifecycle Manager users in any of the LDAP repositories, such as IBM Security Directory Server or Microsoft Active Directory.

About this task

Procedure

1. In the `config.py` properties, update the **`ip`**, **`port`**, **`LDAP_server_type`**, and other properties for your environment. For the description of properties in the `config.py` file, see LDAP integration by using configuration scripts.

   Windows
   `SKLM_INSTALL_HOME\bin\LDAPIntegration\config.py`

   `C:\Program Files\IBM\SKLMV301\bin\LDAPIntegration\config.py`

   Linux `SKLM_INSTALL_HOME/bin/LDAPIntegration/config.py`

   `opt/IBM/SKLMV301/bin/LDAPIntegration/config.py`

   Note: To run the scripts with default configuration, you just need to set the **`ip`** and **`port`** properties.

2. Create the database for LDAP configuration.
   a. Open the DB2 command window.
   b. Run the following command to create the database.

```
db2  create database USERDB31 using codeset UTF-8 territory US
```

3. Update the data source from the WebSphere Integrated Solutions Console with `jndi` name `jdbc/wimXADS`. For the instructions, see "Updating a data source from WebSphere Integrated Solutions Console" on page 199.
4. Create database-based repository to hold all the IBM Security Key Lifecycle Manager application groups.
   a. Go to the <*WAS_HOME*>`\bin` folder.

      Windows
      `C:\Program Files\IBM\WebSphere\AppServer\bin`

      Linux `/opt/IBM/WebSphere/AppServer/bin`
   b. Open a command prompt and run the following commands.

      ```
      wsadmin.bat -user <wasadmin user> -password <wasadmin passwd> -lang jython -f
      <SKLM_INSTALL_HOME>\bin\LDAPIntegration\createDBRepos.py <WAS_HOME> <LDAP_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,

      ```
      LDAP_DBNAME = USERDB31
      SKLM_DBUSER = sklmdb31
      SKLM_DBPORT# = 50050
      ```

      `SKLM_DBUSERPASSWD` is the IBM Security Key Lifecycle Manager database password that you specified during the installation.
5. Run the configuration scripts `sklmLDAPConfigure` and `addLDAPUserToGroup`.

   Windows
   Go to the *SKLM_INSTALL_HOME*`\bin\LDAPIntegration` directory and run the following scripts.

   ```
   sklmLDAPConfigure.bat WAS_HOME SKLM_INSTALL_HOME WAS_ADMIN WASAdmin_PASSWORD SKLM_ADMI
   ```

   ```
   addLDAPUserToGroup.bat WAS_HOME SKLM_INSTALL_HOME WAS_ADMIN WASADMIN_PASS USER_UNIQUE_
   ```

   Linux Go to the *SKLM_INSTALL_HOME*`/bin/LDAPIntegration` directory and run the following script.

   ```
   sklmLDAPConfigure.sh WAS_HOME SKLM_INSTALL_HOME WAS_ADMIN WASAdmin_PASSWORD SKLM_ADMIN
   ```

   ```
   addLDAPUserToGroup.sh WAS_HOME SKLM_INSTALL_HOME WAS_ADMIN WASADMIN_PASS USER_UNIQUE_N
   ```

   WAS_HOME
   The directory where WebSphere Application Server for IBM Security Key Lifecycle Manager is installed.

   Windows
   `drive:\Program Files\IBM\WebSphere\AppServer`

   Linux `path/IBM/WebSphere/AppServer`

   SKLM_INSTALL_HOME
   The directory where IBM Security Key Lifecycle Manager is installed.

   Windows
   `drive``:\Program Files\IBM\SKLMV301`

   Linux

   WAS_ADMIN
   User name of WebSphere Application Server for IBM Security Key Lifecycle Manager.

WAS_PASS
Password of WebSphere Application Server for IBM Security Key Lifecycle Manager.

USER_UNIQUE_NAME
The LDAP user for whom you want to assign IBM Security Key Lifecycle Manager administrator role.

SKLM_ADMIN
Administrator for IBM Security Key Lifecycle Manager.

SKLM_ADMIN_PASS
Password for IBM Security Key Lifecycle Manager administrator.

DB2_install_directory
The directory where DB2 is installed.

Windows
drive:\Program Files\IBM\DB2SKLMV301

Linux
path/IBM/DB2SKLMV301

For non-root installation on Linux , the path is: <*non_root_user_home _directory*>/IBM/DB2SKLMV301

What to do next

After the LDAP configuration, you must run the subsequent tasks. For the details, see "Post-LDAP configuration tasks to support LDAP integration" on page 205

LDAP integration by using configuration scripts

You can run the configuration scripts from a command-line to integrate IBM Security Key Lifecycle Manager with LDAP by using the default configuration settings that are defined in the config.py properties file.

The following example shows the properties that are defined in the config.py file.

```
import string, sys
LDAP_server_type="IDS"
login_id="uid"
ip="9.x.x.x"
port="389"
gr_name="Group"
pr_name="PersonAccount"
gr_obj_class="groupOfUniqueNames"
pr_obj_class="person"
mem_name="uniqueMember"
mem_obj_class="groupOfUniqueNames"
base_entry="o=ibm.com"
scope="direct"
```

The following table provides description for the config.py file properties.

Property	Description
LDAP_server_type	Type of the LDAP server that is being used. By default, IDS is specified.
login_id	Property name that is used for login. For example, uid and mail.
ip	IP address or host name for the primary LDAP server.
port	Port number for the LDAP server.
gr_name	Name of the entity type.

Property	Description
pr_name	Name of the entity type.
gr_obj_class	Object class for the entity type.
pr_obj_class	Object class for the entity type.
mem_name	Name of the LDAP attribute that is used as the group member attribute. For example, `member` or `uniqueMember`.
mem_obj_class	Group object class that contains the member attribute. For example, `groupOfNames` or `groupOfUniqueNames`. If you do not define this parameter, the member attribute applies to all group object classes.
scope	The scope of the member attribute. Specify any of the following values for the parameter. **direct** Member attribute that contains only the direct members. Therefore, this value refers to the member directly contained by the group and not contained through the nested group. For example, if `Group1` contains `Group2` and `Group2` contains `User1`, then `Group2` is a direct member of `Group1` but `User1` is not a direct member of `Group1`. Both `member` and `uniqueMember` are direct member attributes. **nested** Member attribute that contains direct members and the nested members.

If you discover problems during LDAP integration when the scripts are used to run the configuration task, you might need to review the following log files that are at *<SKLM_INSTALL_HOME>*`/bin/LDAPIntegration` to diagnose the problems.

- `sklmldapconf.log`
- `ldaplog.out`

For more information about how to run the configuration scripts, see Running the LDAP configuration scripts.

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 `KLMProfile` folder at `<WAS_HOME>/profiles/KLMProfile`.
      2) Run the **`-validateAndUpdateRegistry`** option of the **`manageProfiles`** command.

         Windows

         ```
         <WAS_HOME>\bin\manageProfiles.bat
         -validateAndUpdateRegistry
         ```

         For example: `C:\Program Files\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\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.ibm.com/support/knowledgecenter/SSEQTP_9.0.0/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 `DB2SKLMV27` folder to the `<`*`WAS_HOME`*`>/lib` folder.

      Default definition of <*WAS_HOME*> variable is typically:

      Windows
      `C:\Program Files\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_PASSWORD>
      -lang jython
      ```
      2) Run the following command:
      ```
      AdminTask.deleteIdMgrDBTables<'[-schemaLocation "<WAS_HOME>/etc/wim/set
      up" -databaseType db2 -dbURL "jdbc:db2://localhost:<SKLMDB_PORT>/
      <LDAPDB_NAME>" -dbDriver com.ibm.db2.jcc.DB2Driver -dbAdminId
      <SKLMDBADMIN_USER> -dbAdminPassword <SKLMDBADMIN_PASSWORD>
      -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 by using WebSphere Integrated Solutions Console" on page 197.
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 by using WebSphere Integrated Solutions Console" on page 197 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:<SKLMDB_PORT>/
      <LDAP_DBNAME>" -dbDriver com.ibm.db2.jcc.DB2Driver -dbAdminId
      <SKLMDB2ADMINUSER> -dbAdminPassword <SKLMDB2ADMINUSER_PASSWORD>
      -reportSqlError true]'>
      ```
   b. Run steps 5 - 9 in "Integrating LDAP by using WebSphere Integrated Solutions Console" on page 197.

Changing DB2 administrator password on LDAP configured servers

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

About this task

If IBM Security Key Lifecycle Manager server is configured with LDAP, you must run the following steps for changing the DB2 administrator password.

Procedure

1. Ensure that the **Allow operations if some of the repositories are down** check box is selected.
   a. Log on to the WebSphere Integrated Solutions Console.
   b. Click **Security** > **Global security**.
   c. In **User account repository** > **Available realm definitions**, select **Federated repositories** from the drop-down list.
   d. Click **Configure**.
   e. Select the **Allow operations if some of the repositories are down** check box.
2. Run the steps that are described in the following topics to change the DB2 administrator password.

   Windows
   Run the steps 3 - 5 in the "Db2 password security issues on Windows systems" on page 33 topic.

   Linux Run the steps 1- 3 in the Db2 password security issues on Linux or AIX systems topic.
3. Update the federated repository settings for the DB repository connection. Also, change the DB2 administrator password by using the **`updateIdMgrDBRepository AdminTask`** command.
   a. Using the **`wsadmin`** interface that the WebSphere Application Server provides, specify the Jython syntax.

      Windows
      ```
      wsadmin.bat -username WASAdmin -password mypwd -lang jython
      ```
      Linux
      ```
      ./wsadmin.sh -username WASAdmin -password mypwd -lang jython
      ```
   b. Run the command, for example:
      ```
      print AdminTask.updateIdMgrDBRepository ('[-id id_name -dbAdminPasswword new_password]')
      ```
      Where `id_name` is `SKLMDBRepos`
      ```
      print AdminConfig.save()
      ```

   For more information, see https://www.ibm.com/support/knowledgecenter/SSAW57_9.0.0/com.ibm.websphere.nd.multiplatform.doc/ae/rxml_atidmgrrepositoryconfig.html#rxml_atidmgrrepositoryconfig__cmd46.
4. Run the steps that are described in the following topics to change the password of the WebSphere Application Server data source.

   Windows
   Run the steps 7 - 8 in the "Db2 password security issues on Windows systems" on page 33 topic.

Linux Run the steps 5 - 6 in the Db2 password security issues on Linux or AIX systems topic.

5. Restart the server. For instructions about how to stop and start the server, see "Restarting the IBM Security Key Lifecycle Manager server" on page 190.

Security standard configurations

You configure IBM Security Key Lifecycle Manager to work with various security standards to meet the specified security requirements for encryption.

Configuring compliance for FIPS in IBM Security Key Lifecycle Manager

You can turn on FIPS for IBM Security Key Lifecycle Manager so that all crypto operations use the IBMJCEFIPS provider, which is FIPS 140-2 certified.

Procedure

1. Set the following property in the *SKLM_HOME*/config/SKLMConfig.properties file.

```
fips=on
```

Command-line interface

a. Go to the <*WAS_HOME*>/bin directory. For example,

Windows

```
cd drive:\Program Files (x86)\IBM\WebSphere\AppServer\
bin
```

Linux

```
cd /opt/IBM/WebSphere/AppServer/bin
```

b. Start the **wsadmin** interface by using an authorized user ID, such as SKLMAdmin. For example,

Windows

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

Linux

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

c. Run the **tklmConfigUpdateEntry** command to set **fips** property in the SKLMConfig.properties configuration file.

```
print AdminTask.tklmConfigUpdateEntry ('[-name fips -value on]')
```

REST interface

a. Open a REST client.

b. 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.

c. Run **Update Config Property REST Service** to set **fips** property in the SKLMConfig.properties configuration file. Pass the user authentication identifier that you obtained in Step b along with the request message as shown in the following example.

```
PUT https://localhost:<port>/SKLM/rest/v1/configProperties
Content-Type: application/json
Accept : application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
Accept-Language : en
{ "fips" : "on"}
```

2. Edit the <*WAS_HOME*>/java_1.7.1_32/jre/lib/security/java.security file and add the IBMJCEFIPS provider to the security providers list as shown in the following example.

   Note: You must update the java.security file for the Java that is used by IBM Security Key Lifecycle Manager server.

   ```
   security.provider.1=com.ibm.crypto.fips.provider.IBMJCEFIPS
   security.provider.2=com.ibm.crypto.provider.IBMJCE
   security.provider.3=com.ibm.jsse.IBMJSSEProvider
   security.provider.4=com.ibm.jsse2.IBMJSSEProvider2
   security.provider.5=com.ibm.security.jgss.IBMJGSSProvider
   security.provider.6=com.ibm.security.cert.IBMCertPath
   security.provider.7=com.ibm.crypto.pkcs11.provider.IBMPKCS11
   security.provider.8=com.ibm.security.cmskeystore.CMSProvider
   security.provider.9=com.ibm.security.jgss.mech.spnego.IBMSPNEGO
   ```

   Note: Providers are renumbered to add the IBMJCEFIPS provider to the top of the list.
3. Save the file.
4. Restart the IBM Security Key Lifecycle Manager server.

Configuring IBM Security Key Lifecycle Manager for Suite B compliance

You can configure IBM Security Key Lifecycle Manager to comply with standards that are specified by the US National Security Agency (NSA) to define security requirements for encryption.

About this task

To enable Suite B compliance in IBM Security Key Lifecycle Manager, you must configure the SKLMConfig.properties properties file with the following option.

```
suiteB=128|192
```

When you configure **suiteB** with the value 128 or 192, the following properties are added to the properties file, or updated, if they already exist.

```
TransportListener.ssl.protocols=TLSv1.2
requireSHA2Signatures=true
autoScaleSignatureHash=true
useThisECKeySize=256(if suiteB is 128)|384(if suiteB is 192)
```

Procedure

1. Set the following property in the *SKLM_HOME*/config/SKLMConfig.properties file.

   ```
   suiteB=128|192
   ```

   - The value 128 specifies the 128-bit minimum level of security.
   - The value 192 specifies the 192-bit minimum level of security.

   Command-line interface

   a. Go to the WAS_HOME/bin directory. For example,

      Windows

      ```
      cd drive:\Program Files\IBM\WebSphere\AppServer\bin
      ```

      Linux

      ```
      cd /opt/IBM/WebSphere/AppServer/bin
      ```

   b. Start the **wsadmin** interface by using an authorized user ID, such as SKLMAdmin. For example,

Windows

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

Linux

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

c. Run the **tklmConfigUpdateEntry** command to set **suiteB** property in the SKLMConfig.properties configuration file.

```
print AdminTask.tklmConfigUpdateEntry ('[-name suiteB -value 128|192]')
```

REST interface

a. Open a REST client.

b. 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.

c. Run **Update Config Property REST Service** to set **suiteB** property in the SKLMConfig.properties configuration file. Pass the user authentication identifier that you obtained in Step b along with the request message as shown in the following example.

```
PUT https://localhost:port/SKLM/rest/v1/configProperties
Content-Type: application/json
Accept : application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
Accept-Language : en
{ "suiteB" : "128|192"}
```

2. Restart the server.

What to do next

Select a certificate that uses the ECDSA algorithm because Suite B compliance requires ECDSA certificate for the SSL communication to work.

If a certificate with the ECDSA algorithm is not available, create a new certificate. For more information, see “Specifying SSL or KMIP certificates” on page 1.

Configuring compliance for NIST SP 800-131A in IBM Security Key Lifecycle Manager

Configure IBM Security Key Lifecycle Manager to communicate over secure sockets in compliance with the National Institute of Standards and Technology (NIST) Special Publication (SP) 800-131A standard in strict mode.

About this task

The NIST SP 800-131A standard specifies algorithms to use to strengthen security and encryption strengths. In strict mode, all communication must conform to SP 800-131A.

This task uses the WASAdmin user ID on the WebSphere Integrated Solutions Console to configure compliance for NIST SP 800-131A in IBM Security Key Lifecycle Manager.

For more information about configuring WebSphere Application Server for SP800-131 standard strict mode, see the IBM WebSphere Application Server documentation (http://www.ibm.com/support/knowledgecenter/en/SSAW57_9.0.0/com.ibm.websphere.nd.multiplatform.doc/ae/tsec_config_strictsp300.html).

Procedure

1. Log on to WebSphere Integrated Solutions Console (`https://localhost:9083/ibm/console/logon.jsp`).
2. On the browser Welcome page, type the user ID `WASAdmin` and the password for this administrator.
3. From the left navigation pane, click **Security** > **SSL certificate and key management**.
4. On the SSL certificate and key management page, under **Related Items**, click **SSL configurations**.
5. Click the **NodeDefaultSSLSettings**.
6. Under the **Additional Properties** section, click **Quality of protection (QoP) settings**.
7. From the **Protocol** list, select **TLSv1.2**, and under the **Cipher suite settings** section, in the **cipher suite groups** section, select **Strong**, and click **Update selected ciphers**.
8. Click **OK**.
9. Click **Save** to save the configuration.
10. From the left navigation pane, click **Security** > **SSL certificate and key management** > **Manage FIPS**.

    To run in a strict SP800-131 mode, all of the certificates that are used for SSL on the server must be converted to certificates that comply with the SP800-131 requirements.
11. To convert certificates, under **Related Items**, click **Convert Certificates**.
12. Select **Strict**, and choose algorithm `SHA256withRSA` to use for the new certificates from the list.
13. Select the size `2048` bits for certificate from the **New certificate key size** list.

    Note: If you choose an Elliptical Curve signature algorithm, they require specific sizes; you are not able to fill in a size. The correct size is used instead.
14. If no certificates are displayed in the **Certificates that can not be converted** frame, click **Apply** and **OK**.

    If certificates are displayed in the **Certificates that can not be converted** frame, server is unable to convert the certificates. Replace these certificates with ones that meet SP800-131 requirements. The server might not convert a certificate for the following reasons:
    - The certificate was created by a Certificate Authority (CA)
    - The certificate is in a read only keystore
15. Click **SSL certificate and key management** > **Manage FIPS**.
16. Select **Enable SP800-131**.
17. Select **Strict**.
18. Click **Apply** and **OK**.
19. Click **Save** to save the configuration.
20. Stop WebSphere Application Server.

    For the steps on how to stop the server, see “Restarting the IBM Security Key Lifecycle Manager server” on page 190.
21. In the `<`*`SKLM_HOME`*`>/config/SKLMConfig.properties` file, set the **`TransportListener.ssl.protocols`** and **`fips`** properties with these values.

    ```
    TransportListener.ssl.protocols=TLSv1.2
    fips=on
    ```

Command-line interface

a. Go to the WAS_HOME/bin directory. For example,

   Windows
   ```
   cd drive:\Program Files\IBM\WebSphere\AppServer\bin
   ```

   Linux `cd /opt/IBM/WebSphere/AppServer/bin`

b. Start the **wsadmin** interface by using an authorized user ID, such as SKLMAdmin. For example,

   Windows
   ```
   wsadmin.bat -username SKLMAdmin -password mypwd -lang jython
   ```

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

c. Run the **tklmConfigUpdateEntry** command.

   ```
   print AdminTask.tklmConfigUpdateEntry ('[-name TransportListener.ssl.protocols -va
   ```

   ```
   print AdminTask.tklmConfigUpdateEntry ('[-name fips -value on]')
   ```

REST interface

a. Open a REST client.

b. 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.

c. Run **Update Config Property REST Service**. Pass the user authentication identifier that you obtained in Step b along with the request message as shown in the following example.

   ```
   PUT https://localhost:<port>/SKLM/rest/v1/configProperties
   Content-Type: application/json
   Accept : application/json
   Authorization: SKLMAuth userAuthId=139aeh34567m
   Accept-Language : en
   { "TransportListener.ssl.protocols" : "TLSv1.2"}
   ```

   ```
   PUT https://localhost:<port>/SKLM/rest/v1/configProperties
   Content-Type: application/json
   Accept : application/json
   Authorization: SKLMAuth userAuthId=139aeh34567m
   Accept-Language : en
   { "fips" : "on"}
   ```

22. In the *<WAS_HOME>*/profiles/KLMProfile/properties/ssl.client.props file, update the properties as shown here.

    ```
    com.ibm.ssl.protocol=TLSv1.2
    com.ibm.security.useFIPS=true
    com.ibm.websphere.security.FIPSLevel=SP800-131
    ```

23. Start WebSphere Application Server.

    For the steps on how to start the server, see "Restarting the IBM Security Key Lifecycle Manager server" on page 190.

Master key management

You can create and refresh IBM Security Key Lifecycle Manager master keys. You can also move a master key from the Java keystore to Hardware Security Module (HSM) and vice versa.

You can use the Master Key REST Service to perform all the master key management operations.

Managing the IBM Security Key Lifecycle Manager master key in a Multi-Master setup

This topic explains the steps to perform the IBM Security Key Lifecycle Manager master key management operations in a Multi-Master cluster. All the master key management operations must be performed on the primary master server only.

Procedure

1. Back up the primary master server. For instructions, see "Backup and restore" on page 113.
2. Ensure that all the master servers in the Multi-master cluster are connected.
3. Perform the master key management operations on the primary master server. For instructions, see Master Key REST Service.
4. On all the master servers, ensure that the value of the **`useMasterKeyinHSM`** property in the `SKLMConfig.properties` file is configured correctly. If the Multi-Master cluster is configured to use HSM, the value of the **`useMasterKeyinHSM`** property must be `true`.

Managing the IBM Security Key Lifecycle Manager master key in a replication setup

This topic explains the steps to perform the master key management operations in a replication setup. After you complete these steps, you can perform the replication of the master server to the clone servers.

Before you begin

Back up the replication master server. For more information, see "Backup and restore" on page 113.

Procedure

1. Perform the master key management operations on the replication master server. For instructions, see Master Key REST Service.
2. Back up the replication master server. For instructions, see "Backup and restore" on page 113.
3. Copy the backup JAR file from the master server to all the clone servers.
4. If the replication master server is configured to use HSM for storing the master key, ensure that all the clone servers are also configured to use the same HSM. For instructions, see "Configuring HSM parameters" on page 194.
5. Restore the backup files on all the clone servers. For instructions, see "Restoring a backup file" on page 124.
6. On all the clone servers, log on to the IBM Security Key Lifecycle Manager graphical user interface as the IBM Security Key Lifecycle Manager administrator.
   a. Click **Administration** > **Replication**.
   b. Select the replication role as **Clone**. The replication role was changed to `Master` after you restored the backup files on the clone server from the master server.
7. Restart the clone servers for which you changed the roles. For instructions, see Restarting the IBM Security Key Lifecycle Manager server.

Multi-Master configuration

Implementation of high-availability solution requires configuration of IBM Security Key Lifecycle Manager masters in a Multi-Master cluster. All IBM Security Key Lifecycle Manager instances in the cluster point to a single data source that is configured for DB2 high availability disaster recovery (HADR) to ensure real-time availability of latest data to all the masters in the cluster.

You can use IBM Security Key Lifecycle Manager Multi-Master configuration for data transmission to achieve the following objectives:

- Ensures consistent and continuous data availability of IBM Security Key Lifecycle Manager across the organization.
- Avoids a single point of failure by using the high-availability solution.
- Masters can be located in several physical sites, that is, distributed across the network.

DB2 high availability disaster recovery (HADR) configuration is used as single data source for all masters in IBM Security Key Lifecycle Manager Multi-Master cluster. HADR protects against data loss by transmitting data changes from a source database, called primary, to a target database, called the standby. DB2 HADR supports multiple standby databases in your Multi-Master setup.

Key features of IBM Security Key Lifecycle Manager Multi-Master configuration

- Keys that are created on an IBM Security Key Lifecycle Manager master are accessible to other IBM Security Key Lifecycle Manager masters in the cluster.
- IPP devices and KMIP clients that are registered on an IBM Security Key Lifecycle Manager master can access keys on another master in the cluster.
- Graphical user interface (GUI) and REST interface to configure IBM Security Key Lifecycle Manager master servers for Multi-Master setup.

For more information about Multi-Master REST services, see Multi-Master configuration REST services.

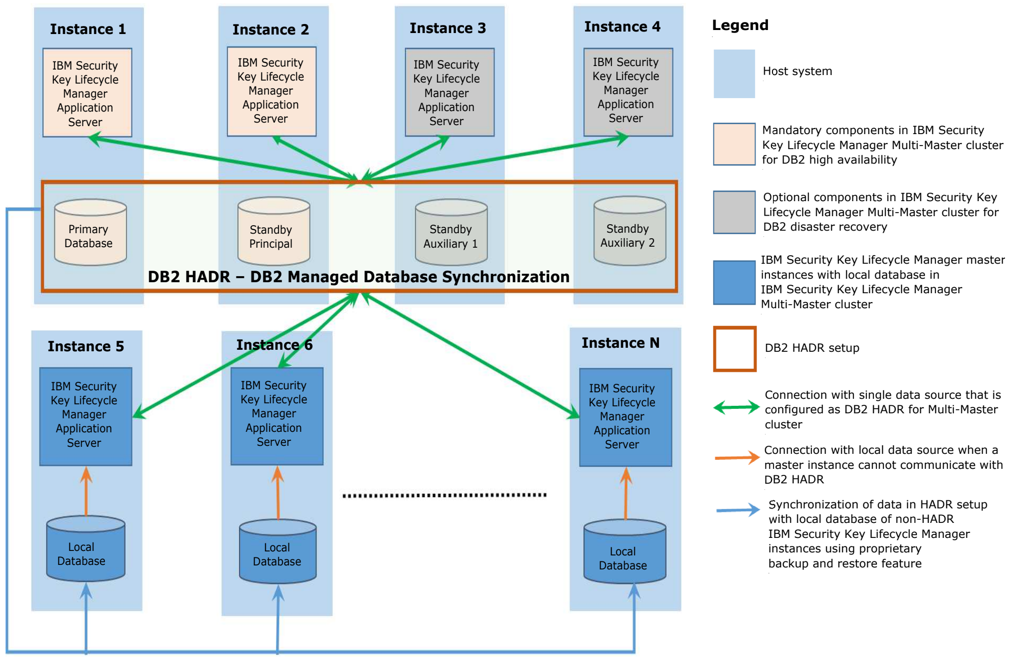
Multi-Master deployment architecture

IBM Security Key Lifecycle Manager Multi-Master architecture is based on DB2 High Availability Disaster Recovery (HADR) feature to implement high-availability solution.

Each instance of IBM Security Key Lifecycle Manager is installed with its DB2 instance that contains all the metadata, key data of managed cryptographic objects, and audit data. The cluster includes multiple instances of IBM Security Key Lifecycle Manager known as masters. All masters in the cluster have same privileges. In the IBM Security Key Lifecycle Manager Multi-Master cluster, each of the masters connects to a single database known as primary database. The primary database connects to another database of an IBM Security Key Lifecycle Manager master known as standby database. With DB2 HADR configuration, data is transmitted continuously between the two databases and synchronized. When the primary database fails, a standby database automatically takes over as the new primary and ensures availability of latest data to all masters in the cluster.

For setting up high availability disaster recovery (HADR), necessary DB2 parameters are configured in IBM Security Key Lifecycle Manager masters with a primary database and a standby database. The following diagrams show a simple deployment of IBM Security Key Lifecycle Manager and DB2 HADR for a Multi-Master environment where four instances (masters) of DB2 HADR and N instances of IBM Security Key Lifecycle Manager are configured.

Physical deployment


Instance 1
IBM Security Key Lifecycle Manager Application Server
Instance 2
IBM Security Key Lifecycle Manager Application Server
Instance 3
IBM Security Key Lifecycle Manager Application Server
Instance 4
IBM Security Key Lifecycle Manager Application Server
Primary Database
Standby Principal
Standby Auxiliary 1
Standby Auxiliary 2
DB2 HADR – DB2 Managed Database Synchronization
Instance 5
IBM Security Key Lifecycle Manager Application Server
Local Database
Instance 6
IBM Security Key Lifecycle Manager Application Server
Local Database
Instance N
IBM Security Key Lifecycle Manager Application Server
Local Database
Legend
Host system
Mandatory components in IBM Security Key Lifecycle Manager Multi-Master cluster for DB2 high availability
Optional components in IBM Security Key Lifecycle Manager Multi-Master cluster for DB2 disaster recovery
IBM Security Key Lifecycle Manager master instances with local database in IBM Security Key Lifecycle Manager Multi-Master cluster
DB2 HADR setup
Connection with single data source that is configured as DB2 HADR for Multi-Master cluster
Connection with local data source when a master instance cannot communicate with DB2 HADR
Synchronization of data in HADR setup with local database of non-HADR IBM Security Key Lifecycle Manager instances using proprietary backup and restore feature

WebSphere Application Server is configured with HADR-enabled DB2 database for automatic client rerouting. When the primary HADR database fails, WebSphere Application Server reestablishes the connection automatically to the principal standby HADR database.

DB2 HADR supports multiple standby databases in your IBM Security Key Lifecycle Manager Multi-Master setup. You can have one principal standby and up to two auxiliary standbys. For more information about multiple standby databases, see Multiple standby databases.

Deployment prerequisites

- Both primary and standby DB2 database servers must be installed on the same version of operating system.
- The DB2 version that is installed on the IBM Security Key Lifecycle Manager primary and standby master servers must match.
- Must use a dedicated network for the DB2 HADR primary and standby connections.

Multiple standby databases

DB2 high availability disaster recovery (HADR) configuration is used to provide continuous data availability to all the IBM Security Key Lifecycle Manager instances in a Multi-Master cluster. HADR protects against data loss by transmitting data changes from a source database, called primary, to a target database, called the standby.

DB2 HADR supports three standby databases in your Multi-Master setup, one standby for high-availability and other two standbys for disaster recovery. When the primary database is down, The HADR takeover service instructs the standby database to take over as a new HADR primary database. For more information about HADR takeover service, see HADR takeover service.

Priorities are assigned to each of the standby database in the cluster. Standby with the higher priority is the one that assumes the primary database role. For example, if a primary database in the IBM Security Key Lifecycle Manager Multi-Master cluster fails, the standby database with a priority index 1 takes over the role of the primary database.

To add multiple standbys to the cluster, you can use IBM Security Key Lifecycle Manager graphical user interface or Add Master REST Service. For more information, see Adding a standby master to Multi-Master cluster.

HADR takeover scenarios

The following table provides information about DB2 HADR takeover scenarios when IBM Security Key Lifecycle Manager masters are configured in a Multi-Master environment.

Primary Database Host System Agent	Primary Database	Standby 1 Database Host System Agent	Standby 1 Database	Standby 2 Database Host System Agent	Standby 2 Database	Standby 3 Database Host System Agent	Standby 3 Database	Agent Actions	IBM Security Key Lifecycle Manager Key Serving	Auto Takeover
Up	Up	Up	Up	Up	Up	Up	Up	Happy scenario.	From primary database	NA
Up	Up	Up	Down	Up or Down	Up or Down	Up or Down	Up or Down	-	From primary database	NA
Up	Up	Down	Down	Up or Down	Up or Down	Up or Down	Up or Down	-	From primary database	NA

Primary Database Host System Agent	Primary Database	Standby 1 Database Host System Agent	Standby 1 Database	Standby 2 Database Host System Agent	Standby 2 Database	Standby 3 Database Host System Agent	Standby 3 Database	Agent Actions	IBM Security Key Lifecycle Manager Key Serving	Auto Takeover
Up	Down	Up	Up	Up or Down	Up or Down	Up or Down	Up or Down	Agent on primary sends request to agent on standby 1 server to takeover as primary database.	From standby 1 database	Yes, if primary or standby 1 database, whichever is available for takeover, and agents on both the servers must be running and can communicate with each other. Else, No.
Down	Down	Up	Up	Up or Down	Up or Down	Up or Down	Up or Down	-	From standby 1 database	No
Up	Down	Up or Down	Down	Up	Up	Up or Down	Up or Down	Agent on primary sends request to agent on standby 2 server to takeover as primary database.	From standby 2 database	Yes, if primary or standby 2 database, whichever is available for takeover, and agents on both the servers must be running and can communicate with each other. Else, No.
Down	Down	Up or Down	Down	Up or Down	Up	Up or Down	Up or Down	-	From standby 2 database after manual takeover	No

Primary Database Host System Agent	Primary Database	Standby 1 Database Host System Agent	Standby 1 Database	Standby 2 Database Host System Agent	Standby 2 Database	Standby 3 Database Host System Agent	Standby 3 Database	Agent Actions	IBM Security Key Lifecycle Manager Key Serving	Auto Takeover
Down	Down	Down	Down	Up or Down	Up	Up or Down	Up or Down	-	From stand by 2 after manual takeover	No
Up	Down	Up or Down	Down	Up or Down	Down	Up	Up	Agent on primary sends request to agent on standby 3 server to takeover as primary database.	From standby 3 database	Yes, if primary or standby 3 database, whichever is available for takeover, and agents on both the servers must be running and can communicate with each other. Else, No.
Up	Down	Up or Down	Down	Up or Down	Down	Up or Down	Down	-	None	No

Monitoring system

In a Multi-Master cluster, monitoring the health status of IBM Security Key Lifecycle Manager instances and quickly correcting problems before they can impact the business operations is essential. IBM Security Key Lifecycle Manager includes monitoring feature to monitor health status of all the IBM Security Key Lifecycle Manager master servers in the cluster.

Monitoring is an integral part of setting up and maintaining your Multi-Master setup. IBM Security Key Lifecycle Manager monitoring system provides a detailed picture of the configuration and health of your Multi-Master environment by using the following monitoring components.

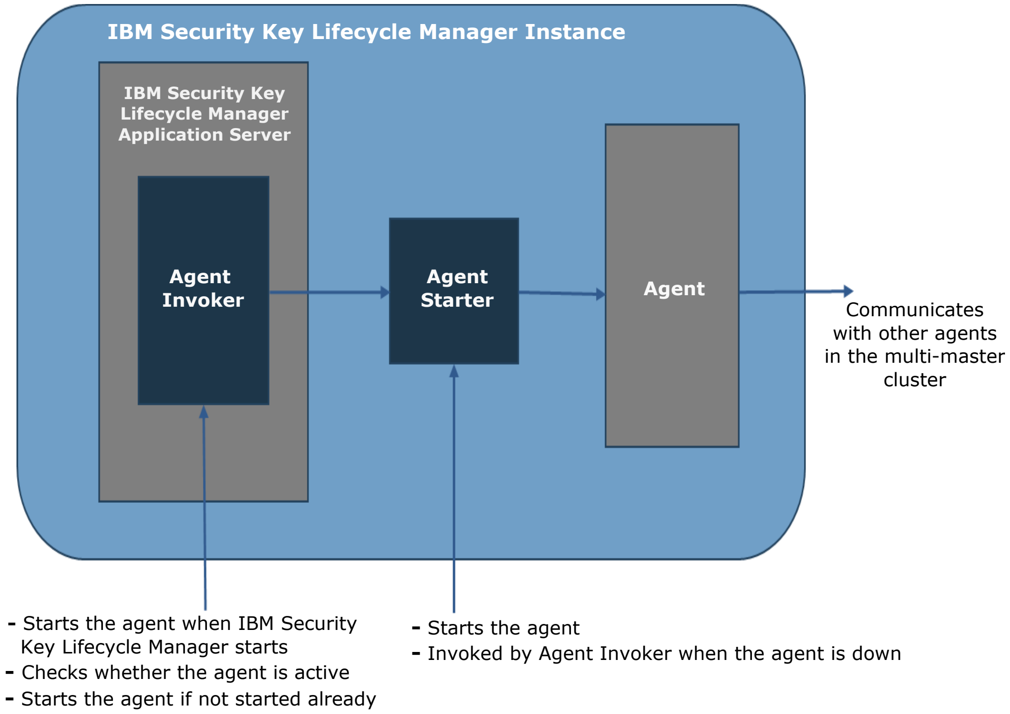
Agent Monitors and collects status data of the IBM Security Key Lifecycle Manager master servers in the cluster at specified intervals. For more information, see Agent.

Agent Starter
Starts the agent service. For more information, see Agent Starter.

Agent Invoker

Monitors status of the agent service at regular intervals. For more information, see Agent Invoker.

Following diagram shows the monitoring components in a IBM Security Key Lifecycle Manager instance (master) of the cluster.


IBM Security Key Lifecycle Manager Instance
IBM Security Key Lifecycle Manager Application Server
Agent Invoker
Agent Starter
Agent
Communicates with other agents in the multi-master cluster
- Starts the agent when IBM Security Key Lifecycle Manager starts
- Checks whether the agent is active
- Starts the agent if not started already
- Starts the agent
- Invoked by Agent Invoker when the agent is down

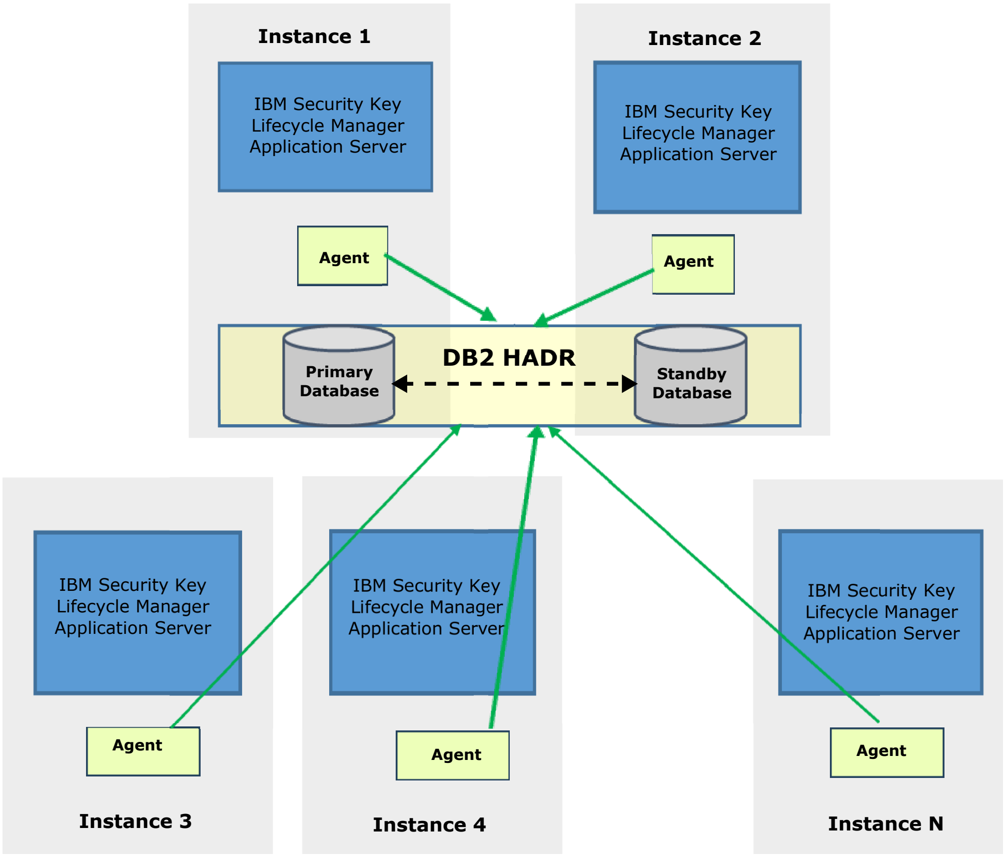
Agent

Use the IBM Security Key Lifecycle Manager agent service to monitor health status and configure IBM Security Key Lifecycle Manager instances in a Multi-Master cluster.

After you install an IBM Security Key Lifecycle Manager instance, the agent is also automatically installed in that server. The agent starts when you start IBM Security Key Lifecycle Manager. When the Agent service is down, the Agent Invoker service runs the Agent Starter script `agentStarter` to restart the service. The `agentStarter.properties` file contains the necessary information to run the script. For more information, see Agent Invoker and Agent Starter.

The following diagram shows how the IBM Security Key Lifecycle Manager agents are deployed in the Multi-Master environment. The agent on each IBM Security Key Lifecycle Manager instance (master) captures the status of UI, KMIP, and IPP ports. Then, the status information is updated in the database with timestamp.

Monitoring Agents


Instance 1
IBM Security Key Lifecycle Manager Application Server
Agent
Instance 2
IBM Security Key Lifecycle Manager Application Server
Agent
DB2 HADR
Primary Database
Standby Database
IBM Security Key Lifecycle Manager Application Server
Agent
Instance 3
IBM Security Key Lifecycle Manager Application Server
Agent
Instance 4
IBM Security Key Lifecycle Manager Application Server
Agent
Instance N

IBM Security Key Lifecycle Manager agent provides the following services to collect monitoring data and configure IBM Security Key Lifecycle Manager instances in the cluster.

Scheduled services

Gathers stratus data by starting and maintaining the following set of services at regular intervals.

- Agent monitoring service
- Port monitoring service
- HADR takeover service

Configuration services

The agent provides several services to set up IBM Security Key Lifecycle Manager masters for Multi-Master configuration. The configuration services are automatically started when the agent is started.

- Configuration services

Agent monitoring service:

The agent monitoring service periodically checks whether agents in the other IBM Security Key Lifecycle Manager master servers of the Multi-Master cluster are up and running.

When the agent in an IBM Security Key Lifecycle Manager master server is started, agent monitoring service automatically starts monitoring the status of agents at regular intervals if IBM Security Key Lifecycle Manager instance is not of type `Local`. You can view the availability status by using IBM Security Key Lifecycle Manager graphical user interface or REST interface. For more information about the agent service, see Agents.

You can use the **`agent.monitoring.svc.interval`** property in the `<`*`SKLM_HOME`*`>\config\SKLMConfig.properties` file, for example, `C:\Program Files\IBM\WebSphere\AppServer\products\sklm\config\SKLMConfig.properties` to configure the agent monitoring service interval. For more information about the configuration property, see agent.monitoring.svc.interval.

For the definition of <*SKLM_HOME*>, see Definitions for *HOME* and other directory variables.

Port monitoring service:

The port monitoring service periodically checks availability of ports that are needed by an IBM Security Key Lifecycle Manager master server for communication in the Multi-Master cluster.

When the agent service in an IBM Security Key Lifecycle Manager master is started, the port monitoring service automatically starts monitoring availability of the ports at regular intervals if IBM Security Key Lifecycle Manager instance is not of type `Local`. You can view the availability status by using IBM Security Key Lifecycle Manager graphical user interface or REST interface. For more information about the agent service, see Agents.

You can use the **`port.monitoring.svc.interval`** property in the `<`*`SKLM_HOME`*`>\config\SKLMConfig.properties` file, for example, `C:\Program Files\IBM\WebSphere\AppServer\products\sklm\config\SKLMConfig.properties` to configure the port monitoring interval. For more information about the configuration property, see port.monitoring.svc.interval.

For the definition of <*SKLM_HOME*>, see Definitions for *HOME* and other directory variables.

The port monitoring service checks whether the following ports are running and reachable.

TCP port
Port on which IBM Security Key Lifecycle Manager server listens for requests from devices.

SSL port
Port on which IBM Security Key Lifecycle Manager server listens for requests from devices that communicate by using the SSL protocol.

KMIP port
Port on which IBM Security Key Lifecycle Manager server listens for requests to communicate over the SSL socket that uses the Key Management Interoperability Protocol (KMIP).

DB2 port
Port on which the IBM Security Key Lifecycle Manager server listens for requests from DB2.

HTTP port
Port on which IBM Security Key Lifecycle Manager listens for HTTPS requests.

Admin port
Port on which the IBM Security Key Lifecycle Manager server listens for requests.

HADR port
Port for the databases that are configured as HADR for database communications.

Agent port
The port on which agent listens for communication from IBM Security Key Lifecycle Manager.

For more information about port default values, see Services, ports, and processes.

HADR takeover service:

The HADR takeover service is responsible to take over from a primary database when a connection problem occurs between IBM Security Key Lifecycle Manager master server and the primary database in the Multi-Master cluster. When the primary database is down, the takeover operation is initiated on a standby database so that the user operations are not hindered during the outage.

You can configure **`agent.takeover.svc.interval`** property in the `<`*`SKLM_HOME`*`>/config/SKLMConfig.properties` file, for example, `C:\Program Files\IBM\WebSphere\AppServer\products\sklm\config\SKLMConfig.properties` to set the time interval for running HADR takeover service. For more information about the configuration property, see agent.takeover.svc.interval.

DB2 High Availability Disaster Recovery (HADR) is used in IBM Security Key Lifecycle Manager Multi-Master cluster. Configuring DB2 HADR protects you against data loss by transmitting data changes from a primary database to standby databases. Under normal conditions, DB2 HADR keeps the DB2 HADR primary and standby databases in sync.

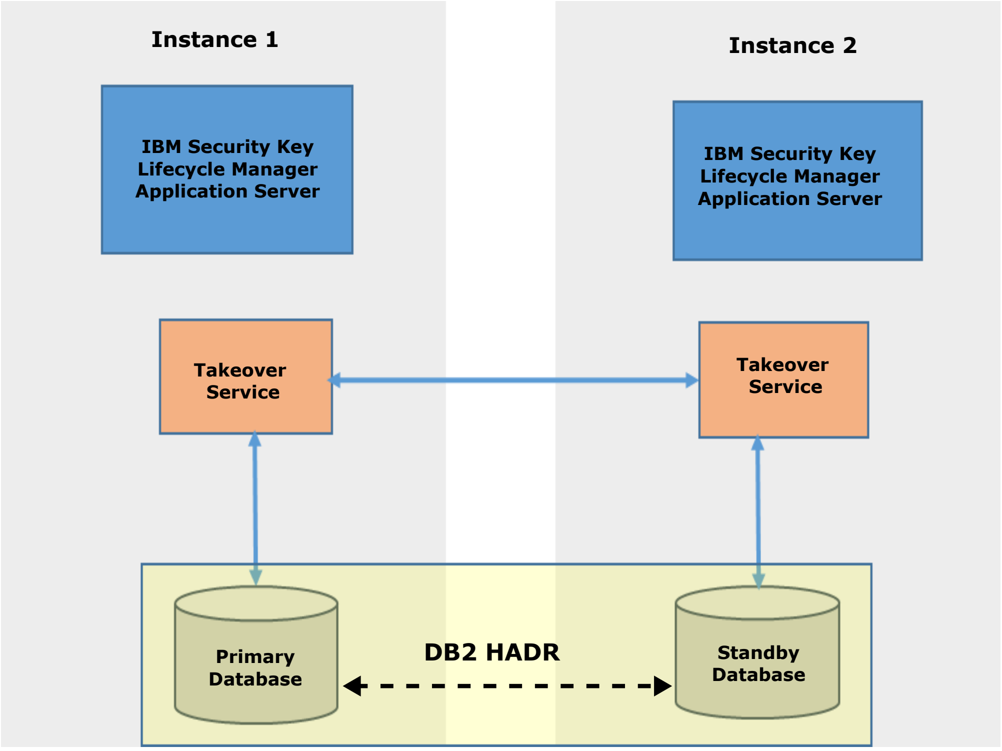
Agents are installed on all the master servers in the cluster. Agent services track the availability of IBM Security Key Lifecycle Manager related ports. If the primary database is down, takeover service instructs the HADR standby database to take over as the new HADR primary database.

For the takeover operation, the primary and standby databases are continuously synchronized by using a secure communication channel. A set of DB2 HADR and WebSphere Application Server configuration parameters are automatically updated for the takeover operation by using the configuration services that the agent runs. For more information about the various configuration services, see Configuration services.

DB2 HADR supports up to three standby databases in your Multi-Master setup. You can have one principal standby and up to two auxiliary standbys. Priorities are assigned to each of the standby database in the cluster. Standby with the higher priority is the one that assumes the primary database role. For example, if a primary database in the IBM Security Key Lifecycle Manager Multi-Master cluster fails, the standby database with a priority index 1 takes over the role as acting primary database. If the takeover operation on standby database with priority index 1 fails, the standby with next priority order (priority index 2) takes over as acting primary database.

Note: You must manually restart WebSphere Application Server in all the standby servers if an axillary standby takes over the primary role. WebSphere Application Server restart is not required when principal standby takes over the primary role.

IBM Security Key Lifecycle Manager supports the failback option. You can configure the primary database to take over the primary role when it comes up.


Instance 1
Instance 2
IBM Security Key Lifecycle Manager Application Server
IBM Security Key Lifecycle Manager Application Server
Takeover Service
Takeover Service
Primary Database
DB2 HADR
Standby Database

- Takeover service of `Instance 1` (primary master server) checks the database status (`Primary Database`) by using DB2 commands.
- If the `Primary Database` is down, `Instance 2` (standby master server) receives takeover request from the primary server. The `Standby Database` takes over as the `Primary Database`.
- The primary master server receives a message from standby to indicate whether the takeover operation is successful. When the takeover operation fails, takeover service on the primary server sends takeover requests to the next standby if the cluster is configured with multiple standby servers.
- When the old primary database server is up, takeover service starts HADR on it as standby.

For more information about prerequisites for DB2 HADR configuration, see Database configuration for high availability disaster recovery (HADR).

Manually initiating takeover operation

When the IBM Security Key Lifecycle Manager primary master server that contains the primary database is down, the takeover operation is not initiated automatically. In such cases, you can manually start the takeover operation by running the **`sklmTakeoverHADR`** script.

Note: If the operating system of the IBM Security Key Lifecycle Manager primary master server fails, use the instructions for manually initiating the takeover operation given here: Operating system of the IBM Security Key Lifecycle Manager primary master server fails.

1. Locate the **`sklmTakeoverHADR`** script.

   Windows

   `<`*`SKLM_INSTALL_HOME`*`>\agent`

   Default location is `C:\Program Files\IBM\SKLMV301\agent`.

   Linux `<`*`SKLM_INSTALL_HOME`*`>/agent`

   Default location is `/opt/IBM/SKLMV301\agent`.

2. Open a command prompt and run the script.

   Windows

   Go to the `<`*`SKLM_INSTALL_HOME`*`>\agent` directory and run the following command:

   ```
   sklmTakeoverHADR.bat <WAS_HOME> [IP_HOSTNAME] [AGENT_PORT]
   ```

   For example,

   ```
   sklmTakeoverHADR.bat "C:\Program Files\IBM\WebSphere\AppServer" 9.113.37.10 60015
   ```

   Linux Go to the `<`*`SKLM_INSTALL_HOME`*`>/agent` directory and run the following command:

   ```
   sklmTakeoverHADR.sh <WAS_HOME> [IP_HOSTNAME] [AGENT_PORT]
   ```

   For example,

   ```
   ./sklmTakeoverHADR.sh /opt/IBM/WebSphere/AppServer 9.113.37.10 60015
   ```

Configuration services:

The agent provides several services to set up IBM Security Key Lifecycle Manager masters for Multi-Master configuration.

Update DB Configuration (Primary)

Updates the primary database in an IBM Security Key Lifecycle Manager master server in the cluster with necessary configurations for Multi-Master setup.

Update HADR DB Configuration (Primary/Standby)

Updates the configuration parameters in both primary and standby database servers for setting up DB2 High Availability Disaster Recovery (HADR).

Take and Restore Backup

Backs up the database from the primary server and restores it on the standby server. In a Multi-Master cluster with DB2 HADR configuration, the primary database server and the standby database server must be synchronized with the same data.

Send and Receive Backup

Sends the backup file from the primary database server to the standby server by using a secure communication channel. On the standby server, the backup file is stored in the `<`*`SKLM_DATA`*`>` folder, for example, `C:\Program Files\IBM\WebSphere\AppServer\products\sklm\data`. For the definition of <*SKLM_DATA*>, see Definitions for *HOME* and other directory variables.

Start HADR (Primary/Standby)
Starts DB2 HADR operations on the primary and standby database servers. Starts DB2 HADR on the standby server, and then on the primary server.

Update WebSphere Application Server Configuration
Updates WebSphere Application Server configuration to specify the DB2 data source properties such as the names and ports of the standby database servers to support automatic client reroute. If the connection to the primary DB2 server fails, WebSphere Application Server reestablishes the connection automatically to the standby DB2 server.

Restart WebSphere Application Server
Restarts WebSphere Application Server on the primary server, standby server, and other IBM Security Key Lifecycle Manager instances in the cluster to apply the DB2 data source configuration changes that are done to support automatic client reroute.

Get DB2 HADR and WebSphere Application Server Setup Status
Gets the DB2 HADR and WebSphere Application Server connection status. For an operating HADR environment, you must ensure that the primary DB2 HADR and the standby DB2 HADR are connected.

Data synchronization service:

In IBM Security Key Lifecycle Manager Multi-Master cluster, the primary and standby databases are configured with DB2 HADR to ensure high-availability. Under normal conditions, DB2 HADR keeps the primary and standby databases in sync. The IBM Security Key Lifecycle Manager data synchronization service copies the DB2 backup files from the primary master to the other master nodes in the cluster at a specified interval. Data synchronization keeps data in the master nodes current with data in the primary server in the cluster.

When a master server is disconnected from the cluster because of connectivity issues, you can set this master server in read-write mode. You can then restore the backup files on the read-write master server to serve keys to the devices. For more information about how to set the isolated master as read-write master, see "Configuring an isolated master as read-write master" on page 243. When connectivity issues are resolved, you can rejoin master to the cluster. For more information about how to rejoin the cluster, see "Rejoining isolated read-write master back to cluster" on page 244.

Backup file location

The backup file from the primary server is copied to the `<WAS_HOME>/products/sklm/data/synchronization` folder on the master node. You can save a maximum of two backup files.

Setting interval for data synchronization

You can configure **`data.synchronizing.svc.interval`** property in the `<SKLM_HOME>/config/SKLMConfig.properties` file to set the time interval for data synchronization. For more information about the configuration property, see data.synchronizing.svc.interval.

Setting password for backup files

You can configure **`data.synchronizing.backup.password`** property in the `<SKLM_HOME>/config/SKLMConfig.properties` file to set password for the backup files that are generated by data synchronization service on the primary or standby master. These backup files are copied to the other master nodes in the IBM Security Key Lifecycle Manager Multi-Master cluster at an interval that you specified for the **data.synchronizing.svc.interval** property.

You can then restore the backup files on the read-write master server by using the password that you set. You can use graphical user interface, command line interface, or REST interface to restore data if you set the password in the configuration file. If the value for the configuration property is not set, a random password is generated and the data is automatically restored on the read-write master. You must restart WebSphere Application Server and the agent service after you set the password. For more information about the configuration property, see data.synchronizing.backup.password.

Setting maximum number of DB2 backup files

You can configure the **data.synchronizing.svc.MaxBackupNum** property in the `<SKLM_HOME>/config/SKLMConfig.properties` file to specify maximum number of DB2 backup files to keep on the non-HADR masters of the Multi-Master cluster. You must restart WebSphere Application Server and the agent service after you set the password. For more information about the configuration property, see data.synchronizing.svc.MaxBackupNum.

Restarting the IBM Security Key Lifecycle Manager agent service:

Restart of the IBM Security Key Lifecycle Manager agent service causes the server to read its configuration and accept the configuration changes, if any.

Procedure

1. Open a command prompt.
2. Go the *SKLM_INSTALL_HOME*\agent directory.

   Windows
   ```
   C:\Program Files\IBM\SKLMV301\agent
   ```

   Linux
   ```
   /opt/IBM/SKLMV301/agent
   ```
3. Stop the agent service by running the following command.

   Windows
   ```
   stopAgent.bat WAS_HOME
   ```
   ```
   stopAgent.bat "C:\Program Files\IBM\WebSphere\AppServer"
   ```

   Linux
   ```
   ./stopAgent.sh <WAS_HOME>
   ```
   ```
   ./stopAgent.sh /opt/IBM/WebSphere/AppServer
   ```
4. Start the agent service by running the following command.

   Windows
   ```
   startAgent.bat WAS_HOME
   ```
   ```
   startAgent.bat "C:\Program Files\IBM\WebSphere\AppServer"
   ```

   Linux
   ```
   ./startAgent.sh WAS_HOME
   ```
   ```
   ./startAgent.sh /opt/IBM/WebSphere/AppServer
   ```

Agent Starter

The Agent Starter service in the IBM Security Key Lifecycle Manager Multi-Master cluster is used to start the monitoring agent.

When the agent in a IBM Security Key Lifecycle Manager master is down, the Agent Invoker service runs the Agent Starter script `startAgent` to restart the service. The `agentStarter.properties` file contains the necessary information to run the script.

Location of the script and the properties file

The `startAgent` script and the `agentStarter.properties` file are in the *SKLM_INSTALL_HOME*`\agent` directory. For example,

Windows

```
C:\Program Files\IBM\SKLMV301\agent\startAgent.bat
```

```
C:\Program Files\IBM\SKLMV301\agent\agentStarter.properties
```

Linux

```
opt/IBM/SKLMV301/agent/startAgent.sh
```

```
opt/IBM/SKLMV301/agent/agentStarter.properties
```

Starting the agent service

Run the following command:

Windows

```
startAgent.bat WAS_HOME
startAgent.bat "C:\Program Files\IBM\WebSphere\AppServer"
```

Linux

```
./startAgent.sh WAS_HOME
./startAgent.sh /opt/IBM/WebSphere/AppServer
```

Sample Agent Starter properties file

```
SELF_DB_PASSWORD=75927941B378990404B33FBD35D3A433
PRIMARY_IP_HOSTNAME=civ3cez161
SERVICE=PortMonitoring,AgentMonitoring,TakeOverService
SELF_IP_HOSTNAME=civ3cez161
SELF_SSL_PORT=441
SELF_DB_NAME=sklmdb31
SELF_INSTANCE_ID=f39dba2
SELF_AGENT_PORT=60015
PRIMARY_DB_PORT=50050
PRIMARY_DB_IP=civ3cez161
SELF_NAME=f39dba2
SELF_SKLM_PASSWORD=A965C364C4DC71657A2A5B1013690045
STANDBY_INSTANCE_COUNT=0
PRIMARY_DB_PASSWORD=75927941B378990404B33FBD35D3A433
SELF_OWNER_EMAIL_ADDR2=
SELF_HTTP_PORT=443
SELF_OWNER_EMAIL_ADDR1=
SELF_WAS_PASSWORD=887A28DD992FC70B894C4BEE509B5876
SELF_SKLM_USERNAME=SKLMAdmin
SELF_HADR_TYPE=1
SELF_DB_PORT=50050
SELF_KEYSTORE_PASSWORD=EDB95C175FCC69347674702DB9C366BC
PRIMARY_DB_USERNAME=sklmdb31
SELF_DB_USERNAME=sklmdb31
SELF_DB_IP=civ3cez161
SELF_KMIP_PORT=5696
SELF_WAS_USERNAME=wasadmin
SELF_TCP_PORT=3801
PRIMARY_AGENT_PORT=60015
SELF_HADR_PORT=60025
NODE_INSTANCE_COUNT=0
```

```
PRIMARY_DB_NAME=SKLMDB31
PRIMARY_HADR_PORT=60025
SELF_ADMIN_PORT=9083
SELF_CLUSTER_NAME=multimaster
```

Possible values for the **SERVICE** parameter are PortMonitoring, AgentMonitoring, TakeOverService, or DataSynchronizeService.

Agent Invoker

The Agent Invoker service in an IBM Security Key Lifecycle Manager master monitors status of the agent at regular intervals.

The Agent Invoker service runs automatically in all IBM Security Key Lifecycle Manager masters of the Multi-Master cluster. When the IBM Security Key Lifecycle Manager application is started, the Agent Invoker service starts checking whether the Agent service is running at regular intervals. If the Agent service is down, the Agent Invoker service restarts agent by using the Agent Starter service.

You can use the **agent.invoker.polling.interval** property in the <*SKLM_HOME*>\config\SKLMConfig.properties file, for example, C:\Program Files\IBM\WebSphere\AppServer\products\sklm\config\SKLMConfig.properties to configure the agent monitoring interval. For more information about the configuration property, see agent.invoker.polling.interval.

Stop Agent

The Stop Agent service in the IBM Security Key Lifecycle Manager Multi-Master cluster is used to stop the monitoring agent.

The stopped agent restarts automatically when the Agent Invoker service is run.

Location of the stopAgent script file

The stopAgent script file is in the *SKLM_INSTALL_HOME*\agent directory. For example,

Windows
C:\Program Files\IBM\SKLMV301\agent\stopAgent.bat

Linux opt/IBM/SKLMV301/agent/stopAgent.sh

Stopping the agent service

Run the following command:

Windows

```
stopAgent.bat WAS_HOME
```

```
stopAgent.bat "C:\Program Files\IBM\WebSphere\AppServer"
```

Linux

```
./stopAgent.sh WAS_HOME
```

```
./stopAgent.sh /opt/IBM/WebSphere/AppServer
```

Stopping the agent permanently

You can stop an agent permanently. When stopped permanently, the agent does not restart automatically when the Agent Invoker service runs.

To stop the agent permanently:

1. In the SKLMConfig.properties file, update the **stopAgentInvocation** property value to true.

   ```
   stopAgentInvocation=true
   ```
2. Stop the agent. See the "Stopping the agent permanently" on page 229 section.

Note: Do not stop the agent permanently on a Multi-Master setup. You can do so on a standalone IBM Security Key Lifecycle Manager server or a replication setup.

To start an agent that was permanently stopped:

1. In the SKLMConfig.properties file, update the value of the **stopAgentInvocation** property to false.

   ```
   stopAgentInvocation=false
   ```

   You can use the tklmConfigUpdateEntry CLI command or the Update Config Property REST Service to update the SKLMConfig.properties file.
2. Restart the WebSphere Application Server.

Requirements and considerations for Multi-Master configuration

Before you set up IBM Security Key Lifecycle Manager Multi-Master environment, review the requirements and considerations to ensure a successful configuration.

- Ensure that the KMIP, SSL, TCP, and agent ports are not blocked for communication before you set up IBM Security Key Lifecycle Manager masters for Multi-Master configuration.
- Ensure that the agent port (60015) and HADR port (60025) that are used for multi-master configuration are not blocked by the firewall.

  Default agent port is 60015, which you can update through UI. Default HADR port is 60025, which is assigned during Multi-Master setup, and that can be configured.
- IBM Security Key Lifecycle Manager Multi-Master architecture is based on Db2 High Availability Disaster Recovery (HADR) technology to implement high-availability solution. Therefore, all the Db2 HADR configuration rules and guidelines are applicable for IBM Security Key Lifecycle Manager Multi-Master configuration.
- Ensure that the IBM Security Key Lifecycle Manager masters with primary and standby Db2 HADR database host systems have the same operating system version and fix pack levels.
- Db2 user name and password must be same on all the masters of IBM Security Key Lifecycle Manager Multi-Master cluster.
- IBM Security Key Lifecycle Manager instance that you want to add to the Multi-Master cluster must not contain any data. Adding of master server with data results in loss of data that was previously created.
  - If you want to add an existing IBM Security Key Lifecycle Manager instance into the cluster, use the device group export and import feature. See "Adding an existing IBM Security Key Lifecycle Manager instance with data to the Multi-Master cluster" on page 236 for more details.
- A TCP/IP interface must be available between primary and standby Db2 HADR database host systems with a dedicated, high speed, and high capacity network bandwidth.
- For IBM Security Key Lifecycle Manager Multi-Master deployment, the cluster must contain a minimum of one primary master and one standby master. When you set up an IBM Security Key Lifecycle Manager Multi-Master cluster, the

server from which you add a master or standby to the cluster becomes the primary master. You must add a standby to the cluster before you add other masters.

- Server certificate must be created in an IBM Security Key Lifecycle Manager instance before you add it to the cluster as the primary master.
- IBM Security Key Lifecycle Manager Multi-Master cluster supports up to three standby masters. When you add standbys to the cluster, priority index value must be in the range of 1-3.
- After the IBM Security Key Lifecycle Manager Multi-Master configuration, you must avoid running the manual backup and restore operations from any of the masters in the cluster.
- Run the IBM Security Key Lifecycle Manager Multi-Master configuration operations only from the primary master of the cluster to avoid any problems.
- Before you add a master to the IBM Security Key Lifecycle Manager Multi-Master cluster on Linux operation system, the permissions for the `/tmp` directory must be set to 777 that is full execute, read, and write permissions.
- Before you add a master server to the cluster, run **`Check Prerequisites REST Service`** to verify whether the master meets all the requirements. For more information about the REST service, see Check Prerequisites REST Service.
- If you want to configure IBM Security Key Lifecycle Manager Multi-Master setup to use HSM to store the master key, you must configure all the masters in the cluster to use the same HSM.
- Before you add a master server to the cluster through the migrated system, you must modify the IBM Security Key Lifecycle Manager administrator user name and the password in the following situations:
  1. When users and groups are migrated from previous version to version 3.0.1 through cross-migration process.
  2. IBM Security Key Lifecycle Manager administrator user name and the password are different than that of the credentials specified during version 3.0.1 installation.
- You cannot remove standby or master server from the Multi-Master cluster if a standby server is down.

IP address to host name mapping

You must ensure that your computer host name is configured correctly before you set up IBM Security Key Lifecycle Manager masters for Multi-Master configuration. You can resolve an IP address to a host name by editing the `etc/hosts` file.

For DB2 HADR configuration, you must update the `etc/hosts` file in the primary and standby master servers of the cluster to enable host name to IP address mapping.

Location of the host file

Windows
`C:\Windows\System32\Drivers\etc\`

Linux `/etc/hosts`

The following example shows the IP address to host name mapping in the `etc/hosts` file.

```
# 127.0.0.1 localhost
# ::1 localhost
9.199.138.209 sklmver3
```

Adding a standby master to the cluster

In IBM Security Key Lifecycle Manager, high-availability solution is implemented by using Multi-Master cluster configuration. IBM Security Key Lifecycle Manager Multi-Master cluster must contain a primary and a standby master. Add a standby master to the cluster for setting up a Multi-Master environment.

Before you begin

Before you add a standby master to the cluster, review the considerations and restrictions that are listed in the Requirements and considerations for Multi-Master configuration topic.

Run Check Prerequisites REST Service to ensure that the master that you want to add meets all requirements and conditions that are defined for IBM Security Key Lifecycle Manager Multi-Master configuration.

About this task

To provide continuous data availability to all the IBM Security Key Lifecycle Manager instances in a Multi-Master cluster, DB2 high-availability disaster recovery (HADR) configuration is used. DB2 HADR is a database replication feature that provides a high-availability solution. HADR protects against data loss by replicating data changes from a source database, called primary, to a target database, called the standby. DB2 HADR supports up to three standby databases in your Multi-Master setup.

When you create an IBM Security Key Lifecycle Manager Multi-Master cluster, the server from which you add a master or standby to the cluster becomes the primary master. Once the cluster is created with a minimum of one primary master and standby master, you can then add masters to the cluster from any of the masters in the cluster. Use the Multi-Master Configuration - Add Master dialog or **`Add Master REST Service`** to add a master to the cluster. Your role must have a permission to add standby master to the IBM Security Key Lifecycle Manager Multi-Master cluster.

You cannot add a standby master to the cluster by using the Multi-Master Configuration - Add Master page when a standby or master server in the cluster is out of network or not reachable. To add a standby master in this scenario, you must use **`Add Master REST Service`** with additional parameters. For more information about the REST service, see REST service for adding a master when other master in the cluster is not reachable.

Procedure

1. Go to the appropriate page or directory.

   Graphical user interface

   a. Log on to the graphical user interface.
   b. On the Welcome page, click **Administration** > **Multi-Master** > **Masters** > **Add Master**.

   REST interface

   Open a REST client.

2. Add a standby master to the cluster.

   Graphical user interface

   a. Click the **Basic Properties** tab.

b. On the Basic Properties dialog, specify information for the standby master that you are adding.

Host name / IP adress	Specify the host name of the IBM Security Key Lifecycle Manager standby master that is added to the cluster.
IBM Security Key Lifecycle Manager user name	Specify the name of the IBM Security Key Lifecycle Manager administrator. The administrator name is displayed by default.
IBM Security Key Lifecycle Manager password	Specify the password for the IBM Security Key Lifecycle Manager server administrator.
WebSphere Application Server user name	Specify the WebSphere Application Server login user ID for the IBM Security Key Lifecycle Manager server administrator profile. The WebSphere Application Server login ID is displayed by default.
WebSphere Application Server password	Specify the password for the WebSphere Application Server login user ID.
UI port	Specify the HTTPS port to access IBM Security Key Lifecycle Manager graphical user interface and REST services. The port number is displayed by default.

c. Click the **Advanced Properties** tab.

d. On the Advanced Properties dialog, specify information for the standby master that you are adding.

Do you want to set this master as standby database?	Select **Yes** to add the current instance of IBM Security Key Lifecycle Manager as a standby master to the cluster.
HADR port	Specify the port number for the standby HADR database to communicate with the primary HADR database.
Standby priority index	Specify the priority index value for the standby database to takeover when the primary database is down. You can set the priority index to any value in the range 1-3. The standby server with a higher priority index level (lower number) takes precedence over the lower-priority databases.

e. Click **Test Connection** to test whether the communication between the standby master that you are adding and the current primary master is successful. For more information, see Perform a test connection.

f. Click **Add**.

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 run **`Add Master 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:<port>/SKLM/rest/v1/ckms/config/nodes/addNodes
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
[
{
"clusterName" : "multimaster",
"hadrPort" : "60020"
},
{
"type" : "Standby",
"ipHostname" :  "cimkc2b151",
"httpPort" :  "443",
"sklmUsername" :  "sklmadmin",
"sklmPassword" :  "SKLM@admin123",
"wasUsername" :  "wasadmin",
"wasPassword" :  "WAS@admin123",
"standbyPriorityIndex" : "1",
"autoAccept" : "Yes"
}
]
```

What to do next

The primary master restarts, and is temporarily unavailable during this process after you add a standby master to the cluster. Verify whether the standby master with its health status information is listed in the Masters table, and also on the IBM Security Key Lifecycle Manager welcome page.

Adding a master to the cluster

In IBM Security Key Lifecycle Manager, high-availability solution is implemented by using Multi-Master cluster configuration. Adding a master to the cluster is part of setting up a Multi-Master environment.

Before you begin

Complete the following tasks:

- Review the considerations and restrictions that are listed in the Requirements and considerations for Multi-Master configuration topic.
- Run Check Prerequisites REST Service to ensure that the master that you want to add meets all requirements and conditions that are defined for IBM Security Key Lifecycle Manager Multi-Master configuration.
- Before you add a non-HADR master to the Multi-Master cluster, ensure that at least one standby master is added in the cluster. For instructions to add a standby master, see “Adding a standby master to the cluster” on page 232.

About this task

When you create an IBM Security Key Lifecycle Manager Multi-Master cluster, the server from which you add a master or standby to the cluster becomes the primary master. Once the cluster is created with a minimum of one primary master and standby master, you can then add masters to the cluster from any of the masters in the cluster. Use the Multi-Master Configuration - Add Master page or **Add Master REST Service** to add a master to the cluster. Your role must have a permission to add master to the IBM Security Key Lifecycle Manager Multi-Master cluster.

You cannot add a master to the cluster by using the Multi-Master Configuration - Add Master page when a standby or master server in the cluster is out of network or not reachable. To add a master in this scenario, you must use **Add Master REST Service** with additional parameters. For more information about the REST service, see REST service for adding a master when other master in the cluster is not reachable.

Procedure

1. Go to the appropriate page or directory.

   Graphical user interface

   a. Log on to the graphical user interface.

   b. On the Welcome page, click **Administration** > **Multi-Master** > **Masters** > **Add Master**.

   REST interface

   Open a REST client.

2. Add a master to the cluster.

   Graphical user interface

   a. Click the **Basic Properties** tab.

   b. On the Basic Properties dialog, specify information for the master that you are adding.

Host name / IP address	Specify the host name of the IBM Security Key Lifecycle Manager instance that is added to the cluster.
IBM Security Key Lifecycle Manager user name	Specify the name of the IBM Security Key Lifecycle Manager administrator. The administrator name is displayed by default.
IBM Security Key Lifecycle Manager password	Specify the password for the IBM Security Key Lifecycle Manager server administrator.
WebSphere Application Server user name	Specify the WebSphere Application Server login user ID for the IBM Security Key Lifecycle Manager server administrator profile. The WebSphere Application Server login ID is displayed by default.
WebSphere Application Server password	Specify the password for the WebSphere Application Server login user ID.
UI port	Specify the HTTPS port to access IBM Security Key Lifecycle Manager graphical user interface and REST services. The port number is displayed by default.

   c. Click **Test Connection** to test whether the communication between the master that you are adding and the current primary master server is successful. For more information, see Perform a test connection.

   d. Click **Add**.

   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 run **`Add Master 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:<port>/SKLM/rest/v1/ckms/config/nodes/addNodes
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
[
{
"clusterName" : "multimaster",
"primaryHadrPort" : "60020"
},
{
"type" : "Node",
"ipHostname": "cimkc2b151",
"httpPort": "443",
"sklmUsername": "sklmadmin",
"sklmPassword": "SKLM@admin123",
"wasUsername": "wasadmin",
"wasPassword": "WAS@admin123",
"autoAccept": "Yes"
}
]
```

What to do next

The primary master restarts, and is temporarily unavailable during this process after you add a master to the cluster. Verify whether the master with its health status information is listed in the Masters table, and also on the IBM Security Key Lifecycle Manager welcome page.

Adding an existing IBM Security Key Lifecycle Manager instance with data to the Multi-Master cluster

You can use the export and import feature of IBM Security Key Lifecycle Manager to add data from an existing IBM Security Key Lifecycle Manager instance to the Multi-Master cluster. You must import the data that was exported from the existing stand-alone instance to the Primary master server that is configured with DB2 HADR.

About this task

You cannot directly add an existing stand-alone instance with data into the cluster. You must first import data from the existing IBM Security Key Lifecycle Manager instance to the primary master. Then, add a master server into the cluster separately.

After data is imported, the data is available on all instances in the cluster. It is up to you to decide whether to add a master separately.

Procedure

1. Export device group data from the existing IBM Security Key Lifecycle Manager instance. For more information about how to export device group data, see "Exporting a device group" on page 107.
2. Import the data that was exported from the existing stand-alone instance to the primary master server that is configured with DB2 HADR. For more information about how to import device group data, see "Importing a device group" on page 108.
3. After you successfully import data to the primary server, you can access data from all the masters in the cluster. If you need a dedicated IBM Security Key Lifecycle Manager master to access the imported data, add a master to the cluster. For more information about adding a master, see "Adding a master to the cluster" on page 234.

What to do next

You might want to decommission the existing stand-alone IBM Security Key Lifecycle Manager instance after you successfully exported the data.

Modifying master details of a cluster

You can change the IBM Security Key Lifecycle Manager multi-master configuration, such as modifying master server details to meet your requirements. For example, you can update the IBM Security Key Lifecycle Manager administrator password.

Before you begin

Before you modify master details of the cluster, review the considerations and restrictions that are listed in the Requirements and considerations for multi-master configuration topic.

About this task

Use the Multi-Master Configuration - Modify Master dialog or **`Modify Master REST Service`** to modify master details.

Your role must have a permission to modify details of a master server in the IBM Security Key Lifecycle Manager multi-master cluster.

Before you add a master server to the cluster through the migrated system, you must modify the IBM Security Key Lifecycle Manager administrator user name and the password in the following situations:

1. When users and groups are migrated from previous version to version 3.0.1 through cross-migration process.
2. IBM Security Key Lifecycle Manager administrator user name and the password are different than that of the credentials specified during version 3.0.1 installation.

Procedure

1. Go to the appropriate page or directory.

   Graphical user interface

   a. Log on to the graphical user interface.
   b. On the Welcome page, click **Administration** > **Multi-Master** > **Masters**.

   REST interface

   Open a REST client.

2. Modify the master details.

   Graphical user interface

   a. From the **Masters** table, select the master that you want to modify.
   b. Click the **Modify Master** tab. Alternatively, double-click the selected master entry.
   c. On the Multi-Master Configuration - Modify Master dialog, modify the master details as necessary.

d. Click **Test Connection** to test the communication between the master that you are modifying and the current primary server is successful. For more information, see Perform a test connection.

e. Click **Update**.

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 run **Modify Master 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:<port>/SKLM/rest/v1/ckms/config/nodes/updateMaster
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
{
"type" : "Standby",
"ipHostname": "cimkc2b151",
"httpPort": "443",
"sklmUsername": "sklmadmin",
"sklmPassword": "SKLM@admin123",
"wasUsername": "wasadmin",
"wasPassword": "WAS@admin123",
}
]
```

What to do next

Verify health status information of the master that you modified on the Masters table, and also on the IBM Security Key Lifecycle Manager welcome page.

Perform a test connection

After you define the parameters for adding or modifying a IBM Security Key Lifecycle Manager master, perform a test connection to ensure that the connection information is correct.

To perform a test connection, click **Test Connection** on the Multi-Master Configuration page. If **Test Connection** returns an error, verify the following settings. Then, test the connection again.

- Verify whether the IBM Security Key Lifecycle Manager master server is reachable.
- Verify whether the host name of the IBM Security Key Lifecycle Manager master server is correct.
- Verify whether the user credentials for IBM Security Key Lifecycle Manager master are correct.
- Verify whether the HTTP port is enabled.

Removing a master from Multi-Master cluster

You can remove a master, which is no longer required in the IBM Security Key Lifecycle Manager Multi-Master cluster.

Before you begin

Before you delete master details of the cluster, review the considerations and restrictions that are listed in the Requirements and considerations for multi-master configuration topic.

About this task

You cannot delete a primary master of the cluster. You can delete a standby master only when the cluster contains multiple standbys. IBM Security Key Lifecycle Manager Multi-Master cluster supports up to three standbys.

Use the IBM Security Key Lifecycle Manager Multi-Master page or **`Remove Master REST Service`** to delete a master.

Your role must have a permission to delete a master of the cluster.

Procedure

1. Go to the appropriate page or directory.

   Graphical user interface

   a. Log on to the graphical user interface.

   b. On the Welcome page, click **Administration** > **Multi-Master**.

   REST interface

   Open a REST client.

2. Delete the master details.

   Graphical user interface

   a. From the **Masters** table, select the master that you want to delete.

   b. Click **Delete Master**.

   c. On the Confirm dialog, read the confirmation message before you delete the master.

   d. Click **OK**.

   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 run **`Remove Master 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:<port>/SKLM/rest/v1/ckms/config/nodes/removeNode
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
[
{"clusterName": "multimaster"},
{"type": "Node",
"ipHostname" : "cimkc2b151",
"httpPort" : "443",
"sklmUsername" : "sklmadmin",
"sklmPassword" : "SKLM@admin123",
"wasUsername" : "wasadmin",
"wasPassword" : "WAS@admin123"}
]
```

3. Restart WebSphere Application Server to refresh the configuration.

What to do next

Verify whether the master that you deleted is removed from the Masters table.

Promoting standby server to primary server

If a primary master in the IBM Security Key Lifecycle Manager multi-master cluster fails, you might want to promote a standby master while you resolve the failure.

Before you begin

Before you promote a standby master in the cluster, review the considerations and restrictions that are listed in the Requirements and considerations for multi-master configuration topic.

About this task

If the primary master becomes unavailable, use the **IBM Security Key Lifecycle Manager Multi-Master** > **HADR Databases** page or `Promote Standby REST Service` to change a standby master to the primary master in the cluster.

Your role must have a permission to change a standby master to the primary master in the IBM Security Key Lifecycle Manager multi-master cluster.

You must manually restart WebSphere Application Server in all the standby servers if an axillary standby is promoted as primary. WebSphere Application Server restart is not required when principal standby is promoted as primary.

Procedure

1. Go to the appropriate page or directory.

   Graphical user interface

   a. Log on to the graphical user interface.
   b. On the Welcome page, click **Administration** > **Multi-Master** > **HADR Databases**.

   REST interface

   Open a REST client.

2. Promote the standby master to primary master server.

   Graphical user interface

   a. From the **HADR Databases** table, select the standby master that you want to promote.
   b. Click **Promote As Primary**.
   c. On the Confirm dialog, read the confirmation message before you promote the standby master.
   d. Click **OK**.

   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 run **Promote Standby 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:<port>/SKLM/rest/v1/ckms/config/nodes/takeoverAsPrimary
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
[
{clusterName:"multimaster"},
{"ipHostname": "civ3cez160"}
]
```

What to do next

Verify role and health status information of the standby master that you promoted on the HADR Databases table, and also on the IBM Security Key Lifecycle Manager welcome page.

Viewing the list of master servers and their configuration status

You can view the list of IBM Security Key Lifecycle Manager master servers and their health status in the multi-master cluster to help you to identify problems, if any, in the masters. You can also view the DB2 HADR configuration status of the primary and standby masters.

About this task

In a multi-master cluster, regularly monitoring the health status of IBM Security Key Lifecycle Manager instances are essential to quickly identify and correct the problems. You can check to see whether all the communication ports are active and reachable on each master server in your multi-master deployment.

Use the IBM Security Key Lifecycle Manager Multi-Master page or **Get All Masters Status REST Service** to view the list of servers and their status.

You can also view the list of masters and status information on the IBM Security Key Lifecycle Manager welcome page.

Procedure

1. Go to the appropriate page or directory.

   Graphical user interface

   a. Log on to the graphical user interface.
   b. On the Welcome page, click **Administration** > **Multi-master**.

   REST interface

   Open a REST client.

2. View the list of servers and their status information to identity problems, if any.

   Graphical user interface

   DB2 HADR configuration status is displayed on the IBM Security Key Lifecycle Manager Multi-Master page.

   a. Click the **Masters** tab to view the list of masters that are configured for multi-master replication, and their configuration status.

b. Click the **HADR Databases** tab to view the list of masters that are configured with DB2 HADR and their configuration status.

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 run **`Get All Masters Status 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.

```
GET https://localhost:<port>/SKLM/rest/v1/ckms/nodes/allNodeStatus
Content-Type: application/json
Accept : application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
Accept-Language : en
```

For more information, see Get All Masters Status REST Service.

What to do next

Use the status information in the table to investigate problems, if any, and to take the necessary actions.

Viewing summary information of a master

Use the Master Details page to view details of a selected master server in the IBM Security Key Lifecycle Manager multi-master cluster for understanding and working with the configuration data.

Procedure

1. Log on to the graphical user interface.
2. Click **Administration** > **Multi-Master** on the Welcome page.
3. Select a master server that is listed in the table.
4. Right-click the master server and then select **Summary**, or double-click the master entry.

   The following table provides the summary information.

IPP	Port on which IBM Security Key Lifecycle Manager server listens for requests from devices that communicate by using the IBM Proprietary Protocol (IPP).
SSL	Port on which IBM Security Key Lifecycle Manager server listens for requests from devices that communicate by using the SSL protocol.
Admin UI	WebSphere Application Server port for the IBM Security Key Lifecycle Manager profile.
Application UI	HTTPS port to access IBM Security Key Lifecycle Manager graphical user interface and REST services.
KMIP	Port on which IBM Security Key Lifecycle Manager server listens for requests to communicate over the SSL socket that uses the Key Management Interoperability Protocol (KMIP).
Database	Port on which the IBM Security Key Lifecycle Manager server listens for requests from DB2.

HADR port	Port for the databases that are configured as HADR for database communications.
Agent	The port on which agent listens for communication from IBM Security Key Lifecycle Manager.

5. Click **OK** to close the summary page.

Configuring an isolated master as read-write master

Due to connectivity/network issues, a master server fails to communicate with other masters in the cluster, and gets isolated from the cluster. You can then configure this isolated master with its local database in read-write mode. After the configuration in read-write mode, all devices and KMIP clients that are registered to the isolated master can communicate seamlessly with the server, without any configuration changes in devices/clients.

Before you begin

Before you set an isolated master as read-write master, review the considerations and restrictions that are listed in the Requirements and considerations for multi-master configuration topic.

About this task

After the configuration in read-write mode, you can add new devices and KMIP clients to the server, and the read-write master can serve them. However, a set of restrictions is imposed on the isolated read-write master on its functions and interfaces. All the delete and modify operations on device groups, key groups, KMIP clients, and managed objects are disabled for data consistency in case the isolated master joins backs the Multi-Master cluster. Administrative functions such as replication, Multi-Master configuration and restore of backup files are also restricted.

Procedure

1. Go to the appropriate page.

   Graphical user interface

   a. Log on to the graphical user interface of isolated master.

   REST interface

   a. Open a REST client.

2. Configure the isolated master as read-write master.

   Graphical user interface

   a. Click the **Set this master as read-write master** link in the notification area on IBM Security Key Lifecycle Manager Welcome page.
   b. On the Confirm dialog, read the confirmation message before you configure the isolated master in read-write mode.
   c. Click **OK**.

   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 run **Set Up Read-Write Master 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:<port>/SKLM/rest/v1/ckms/config/nodes/setupAsReadWriteMaster
Content-Type: application/json
Accept : application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
Accept-Language : en
```

Rejoining isolated read-write master back to cluster

The master server, which was isolated from Multi-Master cluster, and configured as read-write master, can rejoin the cluster when connectivity/network issues are resolved. The data from isolated read-write master is merged to the primary database of the cluster during rejoin process.

Before you begin

Before you rejoin isolated read-write master to the cluster, review the considerations and restrictions that are listed in the Requirements and considerations for multi-master configuration topic.

About this task

The rejoining operation checks for possible conflicts between the data of isolated read-write master and primary master of the cluster. A conflict report is generated to show the conflicts, if any. The isolated read-write master can rejoin the cluster only after all conflicts are resolved. For more information about how to view and resolve the conflicts, see "Viewing conflicts report" on page 245.

Procedure

1. Go to the appropriate page.

   Graphical user interface

   a. Log on to the graphical user interface of isolated master.
   b. Click the **Join back this master to Multi-Master cluster** link in the notification area on the IBM Security Key Lifecycle Manager welcome page.
   c. On the Confirm dialog, read the confirmation message before you rejoin master to the cluster.
   d. Click **OK** to start the rejoin process.

   REST interface

   a. Open a REST client.
   b. 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.
   c. To run **Join Back Cluster REST Service**, send the HTTP GET request. Pass the user authentication identifier that you obtained in Step b along with the request message as shown in the following example.

```
GET https://localhost:<port>/SKLM/rest/v1/ckms/config/nodes/joinBackTheCluster
Content-Type: application/json
Accept : application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
Accept-Language : en
```

2. If any conflicts arise during the rejoin process, the “Conflicts with MM Cluster” window is displayed. See the “Viewing conflicts report” topic for more information.
3. If no data conflicts, the progress dialog box appears. When the process is complete, a message box is displayed to indicate that the rejoin operation is complete.
4. Click **Close**.
5. Restart the server. For instructions about how to stop and start the server, see “Restarting the IBM Security Key Lifecycle Manager server” on page 190.

Viewing conflicts report

During rejoin of isolated read-write master to the cluster, its data is analyzed for conflicts with the data of primary master of the cluster. Conflicts must be resolved before the isolated read-write master data is merged with the primary database. You can view the list of conflicts to analyze and resolve the problems. You can export the conflicts data in comma-separated value (CSV) format.

Procedure

1. Go to the appropriate page.

   Graphical user interface

   a. Log on to the graphical user interface of isolated read-write master.
   b. Click the **Join back this master to Multi-Master cluster** link in the notification area on the IBM Security Key Lifecycle Manager welcome page.
   c. On the Confirm dialog, read the confirmation message before you rejoin master to the cluster.
   d. Click **OK** to run the rejoin process.

   REST interface

   a. Open a REST client.
   b. 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.
   c. To run **`Join Back Cluster REST Service`**, send the HTTP POST request. Pass the user authentication identifier that you obtained in `Step b` along with the request message as shown in the following example.

```
GET https://localhost:<port>/SKLM/rest/v1/ckms/config/nodes/joinBackTheCluster
Content-Type: application/json
Accept : application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
Accept-Language : en
```

2. If any conflicts arise during the rejoin process, a list of conflicts is displayed in the “Conflicts with MM Cluster” window.
3. To export the conflicts data to a file in comma-separated values (CSV) format, click **Export Conflict Report**.

What to do next

You must resolve the conflicts before the isolated read-write master data is merged with the primary database. You can use the following REST services to resolve conflicts.

- **Change Name REST Service**
- **Change Certificate Alias REST Service**
- **Change History REST Service**
- **Renew Key Alias REST Service**

Updating Db2 password in IBM Security Key Lifecycle Manager Multi-Master cluster

When a password expiration restriction is in effect, you must change the password before the expiration period expires.

Before you begin

Ensure that you know the existing password that you want to change.

About this task

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

You must ensure that the Db2 user name and password must be same on all the masters of IBM Security Key Lifecycle Manager Multi-Master cluster.

Procedure

1. Stop Db2 HADR on IBM Security Key Lifecycle Manager primary master with primary database.

   Windows

   a. Click **Start** > **IBM DB2 DBSKLMV301 (Default)** > **DB2 Command Window - Administrator**.

   b. Type the following command and press Enter.

   ```
   db2 stop hadr on database sklmdb31
   ```

   Linux

   a. In a terminal window, type the following command to change the DB2 instance owner.

   ```
   su -sklmdb31
   ```

   b. Run the following command.

   ```
   db2 stop hadr on database sklmdb31
   ```

2. Deactivate standby database on IBM Security Key Lifecycle Manager standby master by running the following command.

   ```
   db2 deactivate db sklmdb31
   ```

3. Stop Db2 HADR on IBM Security Key Lifecycle Manager standby master with standby database by running the following command.

   ```
   db2 stop hadr on database sklmdb31
   ```

4. Stop WebSphere Application Server on all IBM Security Key Lifecycle Manager instances of the multi-master cluster. See the “Restarting the IBM Security Key Lifecycle Manager server” on page 190 topic for the steps.
5. Stop agent service on all IBM Security Key Lifecycle Manager instances. See the “Restarting the IBM Security Key Lifecycle Manager agent service” on page 227 topic for the steps.
6. Stop Db2 on all IBM Security Key Lifecycle Manager servers of the cluster by running the following command.

   ```
   db2stop force
   ```
7. Change the Db2 data source password on all IBM Security Key Lifecycle Manager servers. See the following topic for the steps.

   Windows
   To change password, run the steps 7 - 8 in the following topic.

   Db2 password security issues on Windows systems

   Linux To change password, run the steps 5 - 6 in the following topic.

   “Db2 password security issues on Linux or AIX systems” on page 35
8. Start Db2 on all IBM Security Key Lifecycle Manager servers in the cluster by running the following command.

   ```
   db2start
   ```
9. Start Db2 HADR on all the standby master servers by using the following command.

   ```
   db2 start hadr on database sklmdb31 as standby
   ```
10. Start Db2 HADR on the primary master server by using the following command.

    ```
    db2 start hadr on database sklmdb31 as primary
    ```
11. Start WebSphere Application Server on primary master server. See the “Restarting the IBM Security Key Lifecycle Manager server” on page 190 topic for the steps.
12. Change the Db2 data source password in WebSphere Application Server on the primary master server by using **one** of the following ways:
    - Complete the steps that are given in this technote: http://www.ibm.com/support/docview.wss?uid=ibm10788311

      or
    - Complete the steps as follows:

      Windows
      To change password, run the steps 7 - 8 in the following topic.

      Db2 password security issues on Windows systems

      Linux To change password, run the steps 5 - 6 in the following topic.

      “Db2 password security issues on Linux or AIX systems” on page 35
13. Restart WebSphere Application Server on the primary master server.
14. Update Db2 password in multi-master table by running Update DB2 Password On All Masters REST Service.
15. Stop the agent service on all the master servers. For instructions, see the “Stop Agent” on page 229.
16. Restart WebSphere Application Server on all the non-primary master servers. For instructions, see the “Restarting the IBM Security Key Lifecycle Manager server” on page 190.

17. Change Db2 data source password in WebSphere Application Server on all the non-primary master servers by using one of the following ways:
    - Complete the steps that are given in this technote: http://www.ibm.com/support/docview.wss?uid=ibm10788311

      or
    - Complete the steps as follows:

      Windows
      To change password, run the steps 7 - 8 in the following topic.

      Db2 password security issues on Windows systems

      Linux To change password, run the steps 5 - 6 in the following topic.

      "Db2 password security issues on Linux or AIX systems" on page 35
18. Stop the agent service on all the master servers in the cluster. For instructions, see "Stop Agent" on page 229.
19. Restart WebSphere Application Server on all IBM Security Key Lifecycle Manager servers in the cluster. For instructions, see "Restarting the IBM Security Key Lifecycle Manager server" on page 190

Frequently asked questions about IBM Security Key Lifecycle Manager Multi-Master

Frequently asked questions (FAQs) about IBM Security Key Lifecycle Manager Multi-Master can help you better understand the multi-master configuration processes.

Q) When IBM Security Key Lifecycle Manager masters are configured for multi-master setup, the data synchronization service automatically copies data from the primary server to the master servers at regular intervals. Why do I need to use the Backup and Restore feature of IBM Security Key Lifecycle Manager?

A) As a precautionary measure to avoid possible data loss, use the Backup and Restore feature to back up data manually at regular intervals.

Q) I added a non-HADR IBM Security Key Lifecycle Manager master to the multi-master cluster. Why am I still seeing data from the local database instead of data from the primary database?

A) IBM Security Key Lifecycle Manager data source might still be pointing to local database even after successfully adding the master to the cluster. After adding the master to the cluster, the data source in its WebSphere Application Server must point the primary database.

You can manually update the server details in the data source of the master server that you added by running the following steps.

1. Log on to WebSphere Integrated Solutions Console (https://localhost:9083/ibm/console/logon.jsp).
2. Click **Resources** > **JDBC** > **Data sources** > **SKLM DataSource**.
3. In the SKLM DataSource page, verify the value in the **Server name** field under **Common and required data source properties section**. If the host name of the local server is shown, update the value by specifying the host name of the primary server.

   If the value in the **Server name** field is already updated with the host name of the primary server, restart WebSphere Application Server and close the page. Else, run the following steps.

4. In the **Additional Properties** section of the SKLM DataSource page, click the **WebSphere Application Server data source properties** link.
5. In the **Advanced Db2 features** section, verify the values in the **Alternate server names** field.
6. Specify host name of the standby servers as a comma-separated list in the **Alternate server names** field.
7. Specify port number of the standby servers as a comma-separated list in the **Alternate port numbers** field.
8. Save the changes.
9. Click **Resources** > **JDBC** > **Data sources** > **SKLM scheduler XA Datasource**.
10. Repeat the steps from 3 - 8.
11. Restart WebSphere Application Server.

Q) How do I check that synchronization is running between IBM Security Key Lifecycle Manager primary and standby masters?

A) You can check whether synchronization is running between IBM Security Key Lifecycle Manager primary and standby masters by verifying the time difference between `PRIMARY_LOG_TIME` and `STANDBY_LOG_TIME` Run the following command from Db2 Command Window

```
#db2pd -d <SKLM_DBName> –hadr
```

For example,

```
#db2pd -d sklmdb31 –hadr
```

The following output is displayed.

```
Database Member 0 -- Database SKLMDB31 -- Active -- Up 1 days 21:27:01 -- Date 2018-11-09-

                        HADR_ROLE = PRIMARY
                          REPLAY_TYPE = PHYSICAL
                        HADR_SYNCMODE = SYNC
                           STANDBY_ID = 1
                        LOG_STREAM_ID = 0
                           HADR_STATE = PEER
                           HADR_FLAGS = TCP_PROTOCOL
                  PRIMARY_MEMBER_HOST = WIN-DBA2ALEJOC8
                     PRIMARY_INSTANCE = SKLMDB31
                       PRIMARY_MEMBER = 0
                  STANDBY_MEMBER_HOST = WIN-VB479C09AG3
                     STANDBY_INSTANCE = SKLMDB31
                       STANDBY_MEMBER = 0
                  HADR_CONNECT_STATUS = CONNECTED
             HADR_CONNECT_STATUS_TIME = 11/08/2017 23:25:28.730219 (1510212328)
          HEARTBEAT_INTERVAL(seconds) = 30
                     HEARTBEAT_MISSED = 0
                   HEARTBEAT_EXPECTED = 2490
                HADR_TIMEOUT(seconds) = 120
        TIME_SINCE_LAST_RECV(seconds) = 3
             PEER_WAIT_LIMIT(seconds) = 0
           LOG_HADR_WAIT_CUR(seconds) = 0.000
    LOG_HADR_WAIT_RECENT_AVG(seconds) = 0.001541
   LOG_HADR_WAIT_ACCUMULATED(seconds) = 45.835
                  LOG_HADR_WAIT_COUNT = 19538
SOCK_SEND_BUF_REQUESTED,ACTUAL(bytes) = 0, 65536
SOCK_RECV_BUF_REQUESTED,ACTUAL(bytes) = 0, 65536
            PRIMARY_LOG_FILE,PAGE,POS = S0000003.LOG, 4891, 191226150
            STANDBY_LOG_FILE,PAGE,POS = S0000003.LOG, 4886, 191205494
                  HADR_LOG_GAP(bytes) = 0
     STANDBY_REPLAY_LOG_FILE,PAGE,POS = S0000003.LOG, 4886, 191205494
       STANDBY_RECV_REPLAY_GAP(bytes) = 0
```

```
           PRIMARY_LOG_TIME = 11/09/2017 20:10:52.000000 (1510287052)
           STANDBY_LOG_TIME = 11/09/2017 20:10:20.000000 (1510287020)
    STANDBY_REPLAY_LOG_TIME = 11/09/2017 20:10:20.000000 (1510287020)
STANDBY_RECV_BUF_SIZE(pages) = 4298
    STANDBY_RECV_BUF_PERCENT = 0
  STANDBY_SPOOL_LIMIT(pages) = 380000
       STANDBY_SPOOL_PERCENT = 0
          STANDBY_ERROR_TIME = NULL
        PEER_WINDOW(seconds) = 0
    READS_ON_STANDBY_ENABLED = N
```

In the output, PRIMARY_LOG_TIME shows the time at which the Db2 Transactional logs are updated for Primary server. STANDBY_LOG_TIME shows the time at which the Db2 Transactional logs are updated for standby server. You can ignore the time difference in milliseconds.

Q) How do I view status of ports in the IBM Security Key Lifecycle Manager multi-master setup?

A) IBM Security Key Lifecycle Manager GUI uses icons to represent port status on the multi-master pages. The following table shows the port status icons and their meanings.

Table 5. Status icons and their meanings

Icon	Description
🟩	Port is reachable and serving requests as per the specifications.
❌	Port is not reachable. Service on a specific port might be down. Refresh status by using the Refresh option on the UI page.

Q) How do I view Db2 HADR status in the IBM Security Key Lifecycle Manager multi-master setup?

A) IBM Security Key Lifecycle Manager GUI uses icons to represent Db2 HADR status on the multi-master pages. The following table shows the Db2 HADR status icons and their meanings.

Table 6. Status icons and their meanings

Icon	Description
🟩	Db2 HADR is in running state. All the HADR masters are connected with each other.
⚠	Db2 HADR is in running state, but at least one of the standby HADR masters is not reachable.
❌	Db2 HADR is down and non-functional.

Exporting and importing keys

You can enable data transfer between two IBM Security Key Lifecycle Manager servers by exporting the keys (symmetric or private) from one server (source) and importing them into the other server (target).

Depending on the IBM Security Key Lifecycle Manager version on your server, you can use one of the following methods to export and import keys:

Table 7. Methods to export and import keys

Method	Supported IBM Security Key Lifecycle Manager version	How-to links
Graphical user interface	3.0.1	• "Exporting a key by using the graphical user interface" • "Importing a key by using the graphical user interface" on page 252
REST service	Any supported version.	• Key Export REST Service • Key Import REST Service
CLI command	Any supported version.	• tklmKeyExport • tklmKeyImport

Exporting a key by using the graphical user interface

You can export symmetric and private keys to an encrypted keystore file on an IBM Security Key Lifecycle Manager server. You can then import the keys from this file into another IBM Security Key Lifecycle Manager server to enable data transfer between these servers.

Procedure

1. Go to the appropriate page or directory.
   a. Log on to the graphical user interface.
   b. From the main menu, click **Search**.
   c. In the left Search pane, in **Objects Type**, select **Symmetric Key** or **Private Key**, depending on which keys you want to search. Alternatively, you can also search for device groups whose keys you want to export.
   d. Click **Search**. The keys of selected key type are listed in the right pane.
2. Export the keys to a keystore file.
   a. From the list of keys in the right pane, select the keys that you want to export (Use CTRL to select multiple keys), and click **Export**.
   b. In the Export Symmetric Keys or Export Private Keys window, specify a name for the keystore file that is used to store the exported keys.
   c. Optional: Specify a different file location to save the keystore file. By default, the **File location** field displays the default *`SKLM_DATA`* directory path, where the keystore file is saved. For example, `C:\Program Files\IBM\WebSphere\AppServer\products\sklm\data`. For the definition of *`SKLM_DATA`*, see Definitions for *HOME* and other directory variables.
   d. For symmetric key type: Specify a certificate as the key alias. The Certificate is the public key entry in the keystore that is used to encrypt the symmetric keys. Only the holder of the corresponding private key can access the keys.
   e. For private key type: Create an encryption password. This password will be used to decrypt the keystore file while importing the keys into an IBM Security Key Lifecycle Manager server.
   f. Click **Export**.

What to do next

Import the keys into the IBM Security Key Lifecycle Manager server with which you want to enable data transfer.

Related tasks:

"Importing a key by using the graphical user interface"
You can import symmetric and private keys into an IBM Security Key Lifecycle Manager server to enable data transfer between this server and the server from where the keys were exported. The keys to be imported must be stored in an encrypted keystore file.

Related reference:

Key Export REST Service
Use **`Key Export REST Service`** to export secret keys or public/private key pairs. A secret key is a symmetric key. A public/private key pair is an asymmetric key pair with a public key and a private key.

Importing a key by using the graphical user interface

You can import symmetric and private keys into an IBM Security Key Lifecycle Manager server to enable data transfer between this server and the server from where the keys were exported. The keys to be imported must be stored in an encrypted keystore file.

Before you begin

Copy the keystore file into the default *SKLM_DATA* directory path on the IBM Security Key Lifecycle Manager server where you want to import it. For example, `C:\Program Files\IBM\WebSphere\AppServer\products\sklm\data`. For the definition of *`SKLM_DATA`*, see Definitions for *HOME* and other directory variables.

Procedure

1. Go to the appropriate page or directory.
   a. Log on to the graphical user interface.
   b. On the Welcome page, click **Administration** > **Export and Import** > **Import Keys**.
2. Import a key from a keystore file. You can import one key at a time.
   a. Select the key type.
   b. Click **Browse** to select the keystore file to be imported.
   c. For symmetric key type: Select the certificate to be used to decrypt the keys in the keystore file. Ensure that it is the same certificate that was used when you exported the key.
   d. For private keys: Specify the password to be used to decrypt the keys in the keystore file. Ensure that it is the same password that was used when you exported the keys.
   e. Specify the key alias for the key to be imported.
   f. Optional: To rename the alias of the key that is imported, specify a new alias name. You can rename the alias if the current key alias is already in use or if you want to change it.
   g. Select the device group in which the imported key is to be used.
   h. Click **Import**.
3. To import multiple keys, repeat Step 2 for each key.

Related tasks:

"Importing a key by using the graphical user interface" on page 252
You can import symmetric and private keys into an IBM Security Key Lifecycle Manager server to enable data transfer between this server and the server from where the keys were exported. The keys to be imported must be stored in an encrypted keystore file.

Related reference:

Key Import REST Service
Use **`Key Import REST Service`** to import secret keys or public/private key pairs. A secret key is a symmetric key. A public/private key pair is an asymmetric key pair that contains a public key and a private key. The private key file is in `PKCS#12` format.

Timestamp formats

IBM Security Key Lifecycle Manager supports Coordinated Universal Time (UTC) time syntax.

Following examples shows the IST time stamp in UTC format (GMT + 5:30).

IBM Security Key Lifecycle Manager database
Timestamps are stored in the IBM Security Key Lifecycle Manager database in UTC format.

```
2018-03-22 05:52:07.0
```

User interface
Timestamp values are displayed in the IBM Security Key Lifecycle Manager user interface by using time zone of the browser.

```
Last backup:Mar 22 2018, 05:51:24 AM IST (GMT+05:30)
```

Log files
Timestamp values in all the log files are displayed by using time zone of the server with UTC offset. In the following example, time stamp is shown for IST time zone.

```
Mar 22, 2018 11:31:40 AM +0500
```

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. Go 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
  a. Go to the *<WAS_HOME>*/bin directory. For example,

     Windows
     ```
     cd drive:\Program Files\IBM\WebSphere\AppServer\bin
     ```
     Linux
     ```
     cd /opt/IBM/WebSphere/AppServer/bin
     ```
  b. Start the **wsadmin** interface by using an authorized user ID, such as SKLMAdmin. For example,

     Windows
     ```
     wsadmin.bat -username SKLMAdmin -password mypwd -lang jython
     ```
     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:

     Go 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 30423830303034370000000000000]')
    ```
  - 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 30423830303034370000000000000]')
          ```
       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 30423830303034370000000000000]')
          ```

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 29.

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
   a. Go to the *<WAS_HOME>*/bin directory. For example,

      Windows

      ```
      cd drive:\Program Files\IBM\WebSphere\AppServer\bin
      ```

      Linux

      ```
      cd /opt/IBM/WebSphere/AppServer/bin
      ```

   b. Start the **wsadmin** interface by using an authorized user ID, such as SKLMAdmin. For example,

      Windows

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

      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:<port>/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:<port>/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:<port>/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 = aaa123
     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:<port>/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"
     },
     ]
     ```

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

     Log on to the graphical user interface. The Welcome page is displayed.
   - Command-line interface
     a. Go to the `<WAS_HOME>/bin` directory. For example,

        Windows

        ```
        cd drive:\Program Files\IBM\WebSphere\AppServer\bin
        ```

        Linux

        ```
        cd /opt/IBM/WebSphere/AppServer/bin
        ```

     b. Start the **`wsadmin`** interface by using an authorized user ID, such as `SKLMAdmin`. For example,

        Windows

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

        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. The **File location** field displays the default <*SKLM_DATA*> directory path, where the certificate is exported, for example, `C:\Program Files\IBM\WebSphere\AppServer\products\sklm\data`. For the definition of <*SKLM_DATA*>, see Definitions for *HOME* and other directory variables. Click **Browse** to specify a location under <SKLM_DATA> directory.
     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-a915–8adbfbdca9de
  -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:<port>/SKLM/rest/v1/certificates/export
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
{"uuid":"CERTIFICATE-61f8e7ca-62aa-47d5-a915–8adbfbdca9de",
"format":"DER",
"fileName":"/mycertificate.der"}
```

For more information about **Certificate Export REST Service**, see Certificate Export REST Service.

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:<port>/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:<port>/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, MD-NC119
Armonk, NY 10504-1785
US

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 jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may 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 Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

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. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

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_.

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