Reference

IBM

Contents

Reference

Reference information is organized to help you locate particular facts quickly, such as the description of a certain command-line parameter and REST services.

Reference information is available throughout the documentation. For your convenience, it is gathered and repeated in this Reference topic of the navigation. Several categories of information are provided for quick lookup. For a description of each category of reference information, click the category name in the navigation.

IBM Security Key Lifecycle Manager REST services

The IBM Security Key Lifecycle Manager REpresentational State Transfer (REST) API provides access to the IBM Security Key Lifecycle Manager server functions. You can use HTTP methods to implement the REST architecture-based REST services.

Request format

The IBM Security Key Lifecycle Manager REST API request consists of the following parts:

URL The URL that hosts the RESTful web service.

HTTP method
The REST API uses the following HTTP methods to run various actions on IBM Security Key Lifecycle Manager resources:

GET Lists the specified resource or collection of resources.

POST Creates the specified resource.

PUT Updates or replaces the specified resource.

DELETE
Removes the specified resource.

Request header
The attributes that describe the request to set up the response format.

Request body
More information that is used to process the request. You must pass parameters as JSON (JavaScript Object Notation) object in the request body.

Response format

The REST API service IBM Security Key Lifecycle Manager supports the JSON response format. The RESTful web service responses contain two main components:

Response header
A list of attributes that describes the response format, and includes an HTTP response code.

Response body
The data that represents the resource that you requested or an error message.

Notes:

- For an IBM Security Key Lifecycle Manager REST request message, if you pass duplicate parameters in a JSON request body, the last repeated parameter is processed by the server.
- For an IBM Security Key Lifecycle Manager REST request message, if you pass duplicate parameters as query parameters in the URL, the first repeated parameter is processed by the server.
- All references to the **`alias`** property of cryptographic keys and certificates in the graphical user interface, command-line interface, and REST interface will be deprecated in the later versions of IBM Security Key Lifecycle Manager.

Authentication process for REST services

Before you access IBM Security Key Lifecycle Manager REST services, authenticate to the IBM Security Key Lifecycle Manager server by using your user name and password.

You can use a REST client to access the IBM Security Key Lifecycle Manager REST services. To access a REST service, you must complete the following process:

1. Log in to the IBM Security Key Lifecycle Manager server with your login credentials. You can use "Login REST Service" to access the server. The "Login REST Service" accepts user name and password and returns a unique user authentication identifier.
2. Access the IBM Security Key Lifecycle Manager REST services that provide the required server functions. To access an IBM Security Key Lifecycle Manager REST service, pass the user authentication identifier that you obtained in Step 1 along with the request message.
3. Log out of the IBM Security Key Lifecycle Manager server by using "Logout REST Service" on page 4. To log out, you must pass the user authentication identifier that you obtained in Step 1.

Login REST Service

Use **`Login REST Service`** to log in to the IBM Security Key Lifecycle Manager server with valid user credentials. The REST service validates the credentials and returns a unique user authentication identifier for all subsequent service requests.

Operation
`POST`

URL `https://<host>:<port>/SKLM/rest/v1/ckms/login`

Request

Request Parameters

Parameter	Description
`host`	Specify the IP address or host name of the IBM Security Key Lifecycle Manager server.
`port`	Specify the port number on which the IBM Security Key Lifecycle Manager server listens for requests.

Request headers

Header name	Value
`Content-Type`	`application/json`
`Accept`	`application/json`

Request body

JSON Object with the following specification:

Parameter	Description
userid	Specify the user ID to access the IBM Security Key Lifecycle Manager server.
password	Specify the password that is associated with the user ID.

Response

Response headers

Header name	Value and description
Status Code	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.
Content-Type	`application/json`

Success response body

JSON object with the following specification:

JSON property name	Description
userAuthId	Returns a unique identifier for the authenticated user.

Error Response Body

JSON object with the following specification:

JSON property name	Description
code	Returns the application error code.
message	Returns a message that describes the error.

Examples

Service request for user authentication

```
POST https://localhost:9080/SKLM/rest/v1/ckms/login
Content-Type: application/json
Accept : application/json
{"userid" : "admin1", "password" : "pswd"}
```

Success response

```
Status Code : 200 OK
{"userAuthId" : "37ea1939-1374-4db7-84cd-14e399be2d20"}
```

Error response

```
Status Code : 401 Unauthorised
{"code" : "CTGKM6001E", "message" : "Authentication Failure :
Incorrect user ID/password combination"}
```

Logout REST Service

Use **Logout REST Service** to stop the user session and log out of the IBM Security Key Lifecycle Manager server. The server automatically logs out the user after 15 minutes of inactivity.

Operation

```
DELETE
```

URL `https://<host>:<port>/SKLM/rest/v1/ckms/logout`

Request

Request Parameters

Parameter	Description
host	Specify the IP address or host name of the IBM Security Key Lifecycle Manager server.
port	Specify the port number on which the IBM Security Key Lifecycle Manager server listens for requests.

Request headers

Header name	Value
Content-Type	application/json
Accept	application/json

Request body

JSON Object with the following specification:

JSON property name	Description
userAuthId	Specify the user authentication identifier that you must use to log out from the IBM Security Key Lifecycle Manager server.

Response

Response headers

Header name	Value and description
Status Code	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.

Response headers

Header name	Value and description
Content-Type	application/json

Success response body

JSON object with the following specification:

JSON property name	Description
userId	Returns the user identifier.
logout	Indicates whether the user is logged out of the server. Valid values are `true` or `false`.

Error Response Body

JSON object with the following specification:

JSON property name	Description
code	Returns the application error code.
message	Returns a message that describes the error.

Examples

Service request for user logout

```
DELETE https://localhost:9080/SKLM/v1/ckms/logout
Content-Type: application/json
Accept : application/json
{"userAuthId" : "37ea1939-1374-4db7-84cd-14e399be2d20"}
```

Success response

```
Status Code : 200 OK
{"userid" : "admin","logout" : "true"}
```

Error response

```
Status Code : 400 Bad Request
{"code" : ""CTGKM6002E"", "message" : "Invalid Request: Invalid user
authentication ID or invalid request format"}
```

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.

Note: Some phases of a restore task are brief. You might not observe their return.

Operation
GET

URL https://*<host>*:*<port>*/SKLM/rest/v1/ckms/backupRestore/progress

Request

Request Parameters

Parameter	Description
host	Specify the IP address or host name of the IBM Security Key Lifecycle Manager server.

Request Parameters

Parameter	Description
port	Specify the port number on which the IBM Security Key Lifecycle Manager server listens for requests.

Request Headers

Header name	Value
Content-Type	application/json
Accept	application/json
Authorization	SKLMAuth userAuthId=<authIdValue>
Accept-Language	Any valid locale that is supported by IBM Security Key Lifecycle Manager. For example: en or de

Response

Response Headers

Header name	Value and description
Status Code	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.
Content-Type	application/json
Content-Language	Locale for the response message.

Success response body

JSON object with the following specification:

JSON property name	Description
code	Returns the value that is specified by the **status** property.

Success response body

JSON object with the following specification:

JSON property name	Description
status	Returns any of the following status messages to determine the current phase of a restore task that is running: **0** IDLE **1** INITIALIZE **2** CREATE_TEMP_DIR **3** RESTORE_CONFIG_FILES **4** RESTORE_KEYSTORES **5** RESTORE_DATABASES **6** CLEANUP **7** COMPLETED

Error Response Body

JSON object with the following specification:

JSON property name	Description
code	Returns the application error code.
message	Returns a message that describes the error.

Examples

Service request to determine the current phase of a restore task

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

Success response

```
Status Code : 200 OK
Content-Language: en
{"code":"7","status":"COMPLETED"}
```

Error response

```
Status Code : 400 Bad Request
Content-Language: en
{"code":"CTGKM6002E","message":"CTGKM6002E Bad Request: Invalid user
authentication ID or invalid request format."}
```

Backup Get Restore Result REST Service

Use **Backup Get Restore Result REST Service** to determine the success or failure of a completed restore task.

Operation
GET

URL https://*<host>*:*<port>*/SKLM/rest/v1/ckms/restore/result

Request

Request Parameters

Parameter	Description
host	Specify the IP address or host name of the IBM Security Key Lifecycle Manager server.
port	Specify the port number on which the IBM Security Key Lifecycle Manager server listens for requests.

Request Headers

Header name	Value
Content-Type	`application/json`
Accept	`application/json`
Authorization	`SKLMAuth userAuthId=<authIdValue>`
Accept-Language	Any valid locale that is supported by IBM Security Key Lifecycle Manager. For example: `en` or `de`

Response

Response Headers

Header name	Value and description
Status Code	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.
Content-Type	`application/json`
Content-Language	Locale for the response message.

Success response body

JSON object with the following specification:

JSON property name	Description
code	Returns the value that is specified by the **status** property.
status	Returns the status message to indicate the success or failure of the most recent restore task. **-1** Status of the restore task cannot be determined. The restore task is not run since the last time the IBM Security Key Lifecycle Manager server started. **0** The restore task succeeded. **1** The restore task failed.

Error Response Body

JSON object with the following specification:

JSON property name	Description
code	Returns the application error code.
message	Returns a message that describes the error.

Examples

Service request to determine the success or failure of a completed restore task

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

Success response

```
Status Code : 200 OK
Content-Language: en
{"code":"0","status":"Operation succeeded"}
```

Error response

```
Status Code : 400 Bad Request
Content-Language: en
{"code":"CTGKM6002E","message":"CTGKM6002E Bad Request: Invalid user
authentication ID or invalid request format."}
```

Backup Get Progress REST Service

Use **Backup Get Progress REST Service** to determine the current phase of a backup task that is running.

Operation
`GET`

URL `https://<host>:<port>/SKLM/rest/v1/ckms/backups/progress`

Request

Request Parameters

Parameter	Description
host	Specify the IP address or host name of the IBM Security Key Lifecycle Manager server.
port	Specify the port number on which the IBM Security Key Lifecycle Manager server listens for requests.

Request Headers

Header name	Value
Content-Type	application/json
Accept	application/json
Authorization	SKLMAuth userAuthId=<authIdValue>
Accept-Language	Any valid locale that is supported by IBM Security Key Lifecycle Manager. For example: en or de

Response

Response Headers

Header name	Value and description
Status Code	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.
Content-Type	`application/json`
Content-Language	Locale for the response message.

Success response body

JSON object with the following specification:

JSON property name	Description
code	Returns the value that is specified by the **status** property.
status	Returns any of the following status messages to determine the current phase of a backup task that is running: **0** `IDLE` **1** `INITIALIZE` **2** `BACKUP_CONFIG_FILES` **3** `BACKUP_KEYSTORES` **4** `BACKUP_DATABASE` **5** `CREATE_BACKUP_JAR` **6** `CLEANUP` **7** `COMPLETED`

Error response body

JSON object with the following specification:

JSON property name	Description
code	Returns the application error code.
message	Returns a message that describes the error.

Examples

Service request to determine the current phase of a backup operation that is running

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

Success response

```
Status Code : 200 OK
Content-Language: en
{"code":"7","status":"COMPLETED"}
```

Error response

```
Status Code : 400 Bad Request
Content-Language: en
{"code":"CTGKM6002E","message":"CTGKM6002E Bad Request: Invalid user
authentication ID or invalid request format."}
```

Backup Get Result REST Service

Use **`Backup Get Result REST Service`** to determine the success or failure of the most recent backup task.

Operation

```
GET
```

URL `https://<host>:<port>/SKLM/rest/v1/ckms/backups/result`

Request

Request Parameters

Parameter	Description
`host`	Specify the IP address or host name of the IBM Security Key Lifecycle Manager server.
`port`	Specify the port number on which the IBM Security Key Lifecycle Manager server listens for requests.

Request Headers

Header name	Value
`Content-Type`	`application/json`
`Accept`	`application/json`
`Authorization`	`SKLMAuth userAuthId=<authIdValue>`
`Accept-Language`	Any valid locale that is supported by IBM Security Key Lifecycle Manager. For example: `en` or `de`

Response

Response Headers

Header name	Value and description
`Status Code`	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.
`Content-Type`	`application/json`
`Content-Language`	Locale for the response message.

Success response body

JSON object with the following specification:

JSON property name	Description
`code`	Returns the value that is specified by the **`status`** property.
`status`	Returns the status message to indicates the success or failure of the most recent backup task: **-1** Status of the task cannot be determined. The task is not run since the last time the IBM Security Key Lifecycle Manager server started. **0** The task succeeded. **1** The task failed. **2** Backup operation succeeded with a warning.

Error Response Body

JSON object with the following specification:

JSON property name	Description
`code`	Returns the application error code.
`message`	Returns a message that describes the error.

Examples

Service request to get the result of a most recent backup task

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

Success response

```
Status Code : 200 OK
Content-Language: en
{"code":"0","status":"Operation succeeded"}
```

Error response

```
Status Code : 400 Bad Request
Content-Language: en
{"code":"CTGKM6002E","message":"CTGKM6002E Bad Request: Invalid user
authentication ID or invalid request format."}
```

Backup is Needed REST Service

Use **Backup is Needed REST Service** to determine the keys or certificates in a keystore that are not yet backed up.

Operation
GET

URL `https://<host>:<port>/SKLM/rest/v1/ckms/backups/need`

Request

Request Parameters

Parameter	Description
host	Specify the IP address or host name of the IBM Security Key Lifecycle Manager server.
port	Specify the port number on which the IBM Security Key Lifecycle Manager server listens for requests.

Request Headers

Header name	Value
Content-Type	application/json
Accept	application/json
Authorization	SKLMAuth userAuthId=<authIdValue>
Accept-Language	Any valid locale that is supported by IBM Security Key Lifecycle Manager. For example: en or de

Response

Response Headers

Header name	Value and description
Status Code	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.
Content-Type	application/json
Content-Language	Locale for the response message.

Success response body

JSON object with the following specification:

JSON property name	Description
status	Returns any of the following messages: • CTGKM1304I All keys and certificates have been backed up. • CTGKM1305I There are keys and certificates which have not been backed up. Make a backup.

Error Response Body

JSON object with the following specification:

JSON property name	Description
code	Returns the application error code.
message	Returns a message that describes the error.

Examples

Service request to determine whether there are any keys or certificates that are not yet backed up

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

Success response

```
Status Code : 200 OK
Content-Language: en
{"status":"CTGKM1304I All keys and certificates have been backed up."}
```

Error response

```
Status Code : 400 Bad Request
Content-Language: en
{"code":"CTGKM6002E","message":"CTGKM6002E Bad Request: Invalid user
authentication ID or invalid request format."}
```

Backup List REST Service

Use **Backup List REST Service** to list the IBM Security Key Lifecycle Manager backup files in a directory.

Operation

GET

URL

https://*<host>*:*<port>*/SKLM/rest/v1/ckms/backups

https://*<host>*:*<port>*/SKLM/rest/v1/ckms/backups?backupDirectory=<backupDirectory>

Request

Request Parameters

Parameter	Description
host	Specify the IP address or host name of the IBM Security Key Lifecycle Manager server.

Request Parameters

Parameter	Description
port	Specify the port number on which the IBM Security Key Lifecycle Manager server listens for requests.

Request Headers

Header name	Value
Content-Type	application/json
Accept	application/json
Authorization	SKLMAuth userAuthId=<authIdValue>
Accept-Language	Any valid locale that is supported by IBM Security Key Lifecycle Manager. For example: en or de

Query parameters

Parameter	Description
backupDirectory	Specify a directory that has JAR files. The JAR files contain backup data for IBM Security Key Lifecycle Manager. If you do not specify the path, the path that is specified by the **tklm.backup.dir** property in the SKLMConfig.properties file is appended.

Response

Response Headers

Header name	Value and description
Status Code	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.
Content-Type	application/json
Content-Language	Locale for the response message.

Success response body

JSON object with the following specification:

JSON property name	Description
status	Returns the status message to indicate whether the specified backup file was found.

Success response body

JSON object with the following specification:

JSON property name	Description
productHomeDir	Returns the home directory of IBM Security Key Lifecycle Manager.
backupFilePath	Returns the file path of the backup directory.
timestamp	Returns the time stamp of the last backup.
description	Returns the description of the backup file.
sourceOSType	Returns the operating system of the server on which the backup was created.
sourceOSDescription	Returns the description of the operating system.
backupVersion	Returns the version of the backup file.
checksum	Returns the checksum of the backup file.
createdBy	Returns the name of the user who initiated the backup task.
productVersion	Returns the version of IBM Security Key Lifecycle Manager.
keystoreItems	Returns the details about all keystore file items, which include checksum, backup file path, backup item type, and backup type.
configFileItems	Returns the details about the configuration file items, which include checksum, backup file path, backup item type, and backup type.
databaseMetaDataItem	Returns the details about the database backup, which include checksum, backup file path, backup item type, and backup type.

Error Response Body

JSON object with the following specification:

JSON property name	Description
code	Returns the application error code.
message	Returns a message that describes the error.

Examples

Service request to list the backup files

```
GET https://localhost:9080/SKLM/rest/v1/ckms/backups?backupDirectory=
/sklmbackup
Content-Type: application/json
Accept : application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
Accept-Language : en
```

Success response

```
Status Code : 200 OK
Content-Language: en
[
       {
           "status": "Found 1 backup files. (/tklmbackup)"
       },
       {
           "productHomeDir": "/opt/IBM/WebSphere/AppServer/products/sklm",
           "backupFilePath": "/tklmbackup/./tklm_v3.0.0.0_20130227140515-
```

```
            0600_backup.jar",
           "timestamp": "2013.02.27 20:05:15+0000",
           "description": "Security Key Lifecycle Manager backup",
           "sourceOSType": "UNIX",
           "sourceOSDescription": "Linux:amd64:2.6.27.19-5-default",
           "backupVersion": "1.0",
           "checksum": null,
           "createdBy": "sklmadmin",
            "productVersion": "3.0.0.0 201212181304",
           "keystoreItems": "[ KLMBackupFileItem [ checksum = 949431830
originalFilePath = /opt/IBM/tivoli/tiptklmV2/products/tklm/keystore/
defaultKeyStore jarEntryName = KEYSTORE-79b98f59-857e-4f3d-b2d3-1c4f6a14b663
_defaultKeyStore itemType = KEYSTORE backupType = FILE ], KLMBackupFileItem
[ checksum = 3906517494 originalFilePath = /opt/IBM/WebSphere/AppServer/
products/sklm/keystore/tklmKeystore.jceks jarEntryName = KEY-STORE!
INTERNAL-0ef9a606-5044-437d-a7a5-d2cbb54c4794_tklmKeystore.jceks itemType
= KEYSTORE backupType = FILE ]]",
           "configFileItems": "[ KLMBackupFileItem [ checksum = 4263916783
originalFilePath = /opt/IBM/WebSphere/AppServer/products/sklm/config/
SKLMConfig.properties jarEntryName = CONFIG-9a356148-4936-4143-9b0f-f83e326a
a448_SKLMConfig.properties itemType = CONFIG backupType = FILE ],
KLMBackupFileItem [ checksum = 574882765 originalFilePath = /opt/IBM/
WebSphere/AppServer/products/sklm/config/datastore.properties jarEntryName
= CON-FIG-8c4984b8-8480-450a-867e-7ef3188bc6e9_datastore.properties
itemType = CONFIG backup-Type = FILE ]]",
           "databaseMetaDataItem": " KLMBackupDatabaseItem [ databaseType
= DB2 backupDirec-toryPath = /tmp backupTimestamp = 20130227140515
dbVersion = SQL09072 jarEntryName = META_DATA_ADITI.0.aditi.NODE0000.CATN000
0.20130227140515.001 itemType = META_DATA backupType = DATABASE ]"
       }
    ]
```

Error response

```
Status Code : 400 Bad Request
Content-Language: en
{"code":"CTGKM6002E","message":"CTGKM6002E Bad Request: Invalid user
authenti-cation id or invalid request format."}
```

Backup Run REST Service

Use **Backup Run REST Service** to run the backup task to create backup files. The backup files contain critical data for the current state of the IBM Security Key Lifecycle Manager server.

You can run only one backup or restore task at a time. Ensure that there is sufficient disk space available to contain the backup files.

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_v3.0.0.0_20100123144220-0800_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.

Operation
POST

URL https://*<host>*:*<port>*/rest/v1/ckms/backups

Request

Request Parameters

Parameter	Description
host	Specify the IP address or host name of the IBM Security Key Lifecycle Manager server.
port	Specify the port number on which the IBM Security Key Lifecycle Manager server listens for requests.

Request Headers

Header name	Value
Content-Type	application/json
Accept	application/json
Authorization	SKLMAuth userAuthId=<authIdValue>
Accept-Language	Any valid locale that is supported by IBM Security Key Lifecycle Manager. For example: en or de

Request body

JSON Object with the following specification:

JSON property name	Description
backupDirectory	Optional. Specify the directory where you want to store the JAR files. The JAR files contain backup data for IBM Security Key Lifecycle Manager. You must 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 SKLMgrConfig.properties file. Follow these guidelines to run the backup task: • If you do not specify a value for this parameter and the successful backup is not run, the default directory is SKLM_HOME/backup. • If you specify a relative path such as mybackupdir, the backup is created in the WAS_HOME/profiles/<WASProfile>/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.
databaseBackupDirectory	Optional. Specify a directory in the IBM Security Key Lifecycle Manager database to include temporary backup data for IBM Security Key Lifecycle Manager. If you do not specify the parameter, value of the **tklm.backup.db2.dir** property in the datastore.properties file is used. The file is in the SKLM_HOME\products\sklm\config directory or a temporary system directory if the directory specified by the **tklm.backup.db2.dir** property does not exist. **Note:** Specify the parameter value without a space character.
description	Optional. You can include more information about the purpose or use of the backup file.

Request body

JSON Object with the following specification:

JSON property name	Description
password	Specify a password to encrypt the data in the backup file. The value can range between of 6 and of 32 characters. You can use a different password for each backup file. When you restore a file, you must provide the password that was used to encrypt the file during the backup task.
replica	Optional. The default is n. Specify whether this backup is taken for replication. If you specify y, the replication configuration file is backed up.

Response

Response Headers

Header name	Value and description
Status Code	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.
Content-Type	application/json
Content-Language	Locale for the response message.

Success response body

JSON object with the following specification:

JSON property name	Description
code	Returns the code that is specified by the **status** property.
status	Returns the status message that indicates whether the backup was successful. **-1** State is unknown. The task has not run since the last time the IBM Security Key Lifecycle Manager server started. **0** The backup task succeeded. **1** The backup task failed. **2** Backup task succeeded with a warning.

Error Response Body

JSON object with the following specification:

JSON property name	Description
code	Returns the application error code.
message	Returns a message that describes the error.

Note: **Backup Run REST Service** returns the 500 Internal Server Error response when a backup operation fails. In the response body, you can see only the status message without status code 1.

Examples

Service request to run a backup task

```
POST https://localhost:9080/SKLM/rest/v1/ckms/backups
Content-Type: application/json
Accept : application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
Accept-Language : en
{"backupDirectory":"/sklmbackup","password":"passw0rd"}
```

Success response

```
Status Code : 200 OK
Content-Language: en
{"code":"0","status":"Backup operation succeeded."}
```

Error response

```
Status Code : 400 Bad Request
Content-Language: en
{"code":"CTGKM6002E","message":"CTGKM6002E Bad Request: Invalid user
authentication ID or invalid request format."}
```

Backup Run Restore REST Service

Use **Backup Run Restore REST Service** to restore from an existing backup file. Before you begin, obtain the password that was used to create the backup the file that you intend to use.

Only one backup or restore task can run at a time. Before you start a restore task, isolate the system for maintenance. You must restart the IBM Security Key Lifecycle Manager server immediately after the restore occurs. Verify the environment before you bring the IBM Security Key Lifecycle Manager server back into production.

Operation

POST

URL https://*<host>*:*<port>*/SKLM/rest/v1/ckms/restore

Request

Request Parameters

Parameter	Description
host	Specify the IP address or host name of the IBM Security Key Lifecycle Manager server.
port	Specify the port number on which the IBM Security Key Lifecycle Manager server listens for requests.

Request Headers

Header name	Value
Content-Type	application/json
Accept	application/json
Authorization	SKLMAuth userAuthId=<authIdValue>
Accept-Language	Any valid locale that is supported by IBM Security Key Lifecycle Manager. For example: en or de

Request body

JSON Object with the following specification:

JSON property name	Description
backupFilePath	Specify the full path and file name that contains backup data. To determine this directory, examine the value of the **sklm.backup.dir** property in the SKLMgrConfig.properties file.
password	Specify a password to encrypt the data in the backup file that you want to restore. You can use a different password for each backup file. When you restore a file, you must provide the password that was used to encrypt the data in that file during the backup task.

Response

Response Headers

Header name	Value and description
Status Code	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.
Content-Type	application/json
Content-Language	Locale for the response message.

Success response body

JSON object with the following specification:

JSON property name	Description
code	Returns the value that is specified by the **status** property.

Success response body

JSON object with the following specification:

JSON property name	Description
status	Returns the status message to indicate whether the restore task was successful. **-1** State is unknown. The task is not run since the last time the IBM Security Key Lifecycle Manager server started. **0** The restore succeeded. **1** The restore task failed.

Error Response Body

JSON object with the following specification:

JSON property name	Description
code	Returns the application error code.
message	Returns a message that describes the error.

Note: **Backup Run Restore REST Service** returns the 500 Internal Server Error response when a restore operation fails. In the response body, you can see only the status message without status code 1.

Examples

Service request to run a backup restore task

```
POST https://localhost:9080/SKLM/rest/v1/ckms/restore
Content-Type: application/json
Accept : application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
Accept-Language : en
{"backupFilePath":"/sklmbackup","password":"passw0rd"}
```

Success response

```
Status Code : 200 OK
Content-Language: en
{"code":"0","status":"Restore operation succeeded. Restart the server."}
```

Error response

```
Status Code : 400 Bad Request
Content-Language: en
{"code":"CTGKM6002E","message":"CTGKM6002E Bad Request: Invalid user
authentication ID or invalid request format."}
```

Certificate Attribute Update REST Service

Use **Certificate Attribute Update REST Service** to update certificate metadata that are Key Management Interoperability Protocol attributes in the database.

Operation

PUT

URL https://*<host>*:*<port>*/SKLM/rest/v1/certificateAttributes

Request

Request Parameters

Parameter	Description
host	Specify the IP address or host name of the IBM Security Key Lifecycle Manager server.
port	Specify the port number on which the IBM Security Key Lifecycle Manager server listens for requests.

Request Headers

Header name	Value
Content-Type	application/json
Accept	application/json
Authorization	SKLMAuth userAuthId=<authIdValue>
Accept-Language	Any valid locale that is supported by IBM Security Key Lifecycle Manager. For example: en or de

Request body

JSON object with the following specification:

JSON property name	Description
`attrName`	Required. Specify the name that you can use to identify or locate the attribute pair as an object. **Note:** Do not use an asterisk (*) or question mark (?) as a character in a Key Management Interoperability Protocol attribute. These wildcard characters are reserved for future use.You can specify the following attributes: **applicationSpecificInformation** Specifies application namespace information as a Key Management Interoperability Protocol attribute. **contactInformation** Specifies contact information as a Key Management Interoperability Protocol attribute. **cryptoParams cryptoparameter1, cryptoparameterN** Specifies the cryptographic parameters that you use for cryptographic operations by using the object. This attribute is a Key Management Interoperability Protocol attribute. **customAttribute** Specifies a custom attribute in string format as a Key Management Interoperability Protocol attribute. Client-specific attributes must start with the characters "x-" (x hyphen) and server-specific attributes must start with "y-" (y hyphen). **link** Specifies the link from one managed cryptographic object to another, closely related target managed cryptographic object. This attribute is a Key Management Interoperability Protocol attribute. **name** Specifies the name that you use to identify or locate the object. This attribute is a Key Management Interoperability Protocol attribute. **objectGroup** Specifies one or more object group names of which this object might be part. This attribute is a Key Management Interoperability Protocol attribute.

Request body

JSON object with the following specification:

JSON property name	Description
`attrValue`	Conditional. Specify one or more of these key value pairs to add or update: **applicationSpecificInformation applicationIDstring** Specifies application namespace information as the value of `applicationIDstring`. **NAMESPACE** Application namespace. **INFO** Application namespace information. **contactInformation contactstring** Specifies contact information as the value of `contactstring`. This attribute is a Key Management Interoperability Protocol attribute. **VALUE** Contact information.
	cryptoParams cryptoparameter1, cryptoparameterN Specifies the cryptographic parameters that you use for cryptographic operations by using the object. This attribute is a Key Management Interoperability Protocol attribute. **MODE** CBC, ECB, PCBC, CFB, OFB, CTR, CMAC, CCM, GCM, CBC_MAC, XTS, AES_KEY_WRAP_PADDING, NIST_KEY_WRAP, X9_102_AESKW, X9_102_TDKW, X9_102_AKW1, X 9_102_AKW2 **PAD** NONE, OAEP, PKCS5, SSL3, ZEROS, ANSI_X9_23, ISO_10126, PKCS1_ V1_5, X9_31, PSS **HASH** MD2, MD4, MD5, SHA1, SHA224, SHA256, SHA384, SHA512 **ROLE** BDK, CVK, DEK, MKAC, MKSMC, MKSMI, MKDAC, MKDN, MKCP, MKOTH, KEK, MAC1660 9, MAC97971, MAC97972, MAC97973, MAC97974, MAC97975, ZPK, PVKIBM, PVKPVV, PVKOTH

Request body

JSON object with the following specification:

JSON property name	Description
	customAttribute customstring Specifies for the value of **`customstring`** a custom attribute in string format as a Key Management Interoperability Protocol attribute. Client-specific attributes must start with the characters "x-" (x hyphen) and server-specific attributes must start with "y-" (y hyphen). **NAME** Client or server attribute name. **VALUE** Value of the attribute name.
	link objectname, objectnametarget Specifies the link from one managed cryptographic object to another, closely related target managed cryptographic object. This attribute is a Key Management Interoperability Protocol attribute. **TYPE** CERTIFICATE, PRIVATE_KEY, PUBLIC_KEY, DERIVATION_BASE_OBJECT, DERIVED_KEY, REPLACEMENT_OBJECT, REPLACED_OBJECT **LINKED_OBJECT_ID** Specify the target uuid of the linked object.
	name Specifies the name that you to identify or locate the object. This attribute is a Key Management Interoperability Protocol attribute. **TYPE** TEXT, URI **VALUE** Name, or URI identifying the object.
	objectGroup objectgroupname1, objectgroupnameN Specifies for **`objectgroupname1`**, **`objectgroupnameN`** the values of one or more object group names of which this object might be part. This attribute is a Key Management Interoperability Protocol attribute. **VALUE** Object group name.
`index`	Conditional. Specify the index to update or delete an attribute value.
`operation`	Required. Specify one of these valid operations to run on an attribute value: `add`, `update`, or `delete`
`uuid`	Required. Specify the universal unique identifier of the certificate.

Response

Response Headers

Header name	Value and description
Status Code	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.
Content-Type	`application/json`
Content-Language	Locale for the response message.

Success response body

JSON object with the following specification:

JSON property name	Description
code	Returns the code that is specified by the **status** property.
status	Returns the status to indicate whether the certificate attribute update task is successful.

Error Response Body

JSON object with the following specification:

JSON property name	Description
code	Returns the application error code.
message	Returns a message that describes the error.

Examples

Service request to add an attribute to a certificate

```
PUT https://localhost:9080/SKLM/rest/v1/certificateAttributes
Content-Type: application/json
Accept : application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
{"uuid":"CERTIFICATE-d3ee4491-f96e-495d-bb37-fc03748924ba","operation":
"add","attrName":"cryptoParams","attrValue":"MODE CBC, PAD NONE,HASH
SHA256,ROLE BDK"}
```

Success response

```
Status Code : 200 OK
{"code": "0","status": "Succeeded"}
```

Service request to add an attribute for a certificate name

```
PUT https://localhost:9080/SKLM/rest/v1/certificateAttributes
Content-Type: application/json
Accept : application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
{"uuid":"CERTIFICATE-d3ee4491-f96e-495d-bb37-fc03748924ba","operation":
"add","attrName":"name","attrValue":"TYPE TEXT,VALUE cert name for xyz"
```

Success response

```
Status Code : 200 OK
{"code": "0","status": "Succeeded"}
```

Service request to update an attribute

```
PUT https://localhost:9080/SKLM/rest/v1/certificateAttributes
Content-Type: application/json
Accept : application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
{"uuid":"KEY-d3ee4491-f96e-495d-bb37-fc03748924ba","operation":"update",
"index":"0","attrName":"name","attrValue":"TYPE TEXT,VALUE updated cert
name for xyz"}
```

Success response

```
Status Code : 200 OK
{"code": "0","status": "Succeeded"}
```

Service request to delete an attribute

```
PUT https://localhost:9080/SKLM/rest/v1/certificateAttributes
Content-Type: application/json
Accept : application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
{"uuid":"KEY-d3ee4491-f96e-495d-bb37-fc03748924ba","operation":"delete",
"index":"0","attrName":"name"}
```

Success response

```
Status Code : 200 OK
{"code": "0","status": "Succeeded"}
```

Service request to update an attribute when an invalid parameter is specified

```
PUT https://localhost:9080/SKLM/rest/v1/certificateAttributes
Content-Type: application/json
Accept : application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
{"UUID":"CERTIFICATE-d3ee4491-f96e-495d-bb37-fc03748924ba","operation":
"add","attrName":"cryptoParams","attrValue":"MODE CBC, PAD NONE,HASH
SHA256,ROLE BDK"}
```

Error response

```
Status Code : 400 Bad Request
{"code":"CTGKM0630E","message":"CTGKM0630E Validation error:
 \"Invalid name \" for parameter \"UUID\"."}
```

Create Certificate REST Service

Use **`Create Certificate REST Service`** to create a certificate and a public and private key pair. The newly created certificate is stored in an existing keystore.

Operation
POST

URL `https://<host>:<port>/SKLM/rest/v1/certificates`

Request

Request Parameters

Parameter	Description
host	Specify the IP address or host name of the IBM Security Key Lifecycle Manager server.
port	Specify the port number on which the IBM Security Key Lifecycle Manager server listens for requests.

Request Headers

Header name	Value
Content-Type	application/json
Accept	application/json
Authorization	SKLMAuth userAuthId=<authIdValue>
Accept-Language	Any valid locale that is supported by IBM Security Key Lifecycle Manager. For example: en or de

Request body

JSON Object with the following specification:

JSON property name	Description
type	Required. Specify the certificate type. The supported certificate type is Self-signed. The subject name and issuer name of the certificate are same.
alias	Required. Specify a unique name for the certificate. The name is not case-sensitive. For example, if you specify MY Cert1, the value is stored as my cert1. Do not use: • The value that begins with 3 alphabetic characters followed by 18 numeric characters, such as aaa000000000000000002. IBM Security Key Lifecycle Manager uses this format to generate a key group with symmetric keys. • Forward slash (/) or backslash (\) characters in the value.
cn	Required. Specify a common name for the certificate that you want to create.
ou	Specify the organizational unit name.
o	Specify the organizational name.
country	Specify the country name. Indicate the name as a two-letter country code.

Request body

JSON Object with the following specification:

JSON property name	Description
usage	Required. Specify the target application usage with the following values: **3592** Specifies the 3592 device group. **DS8000** Specifies the DS8000 device group. **GENERIC** Specifies a device family that uses the Key Management Interoperability Protocol to interact with IBM Security Key Lifecycle Manager. The GENERIC device group enables management of KMIP objects. **SSLCLIENT** Specifies the client-side certificate that is used in secure communication by using Secure Socket Layer protocol to authenticate the client device. **SSLSERVER** Specifies the server-side certificate that is used in secure communication by using Secure Socket Layer protocol. **userdevicegroup** Specifies a user-defined group that is based on a supported device family.
validity	Required. Specify the days during which the certificate is valid. The interval can range from 1 day to 9000 days.
algorithm	Required. Specify any of the following cryptographic algorithms that the certificate can use. • RSA • ECDSA

Response

Response Headers

Header name	Value and description
Status Code	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.
Content-Type	application/json
Content-Language	Locale for the response message.

Success response body

JSON object with the following specification:

JSON property name	Description
status	Returns the status to indicate the certificate creation.

Error Response Body

JSON object with the following specification:

JSON property name	Description
code	Returns the application error code.
message	Returns a message that describes the error.

Examples

Service request to create a certificate

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

Success response

```
Status Code : 200 OK
Content-Language: en
{"Status":"Created a key pair and self-signed certificate: sklmCertificate"}
```

Error response

```
Status Code : 400 Bad Request
Content-Language: en
{"code":"CTGKM6002E","message":"CTGKM6002E Bad Request: Invalid user
authentication ID or invalid request format."}
```

```
Status Code : 500 Internal Server Error
Content-Language: en
{"code":"CTGKM0540E","message":"CTGKM0540E Certificate with alias
sklmCertificate already exists in keystore."}
```

Certificate Generate Request REST Service

Use **Certificate Generate Request REST Service** to create a PKCS #10 certificate request file. This service creates certificate request file, such as SKLM_HOME\080419154137–sslcert001.csr. You must manually send the request to a certificate authority.

When the certificate authority returns a certificate in response to this request, copy the certificate to a file. Use **Certificate Import REST Service** to load the response file. You must specify the same alias name that was used with **Certificate Generate Request REST Service** to generate the request.

After you generate the certificate request, the certificate activation date and creation date are identical. This certificate is available to the key server and drive.

Operation

POST

URL `https://<host>:<port>/SKLM/rest/v1/certificates`

Request

Request Parameters

Parameter	Description
host	Specify the IP address or host name of the IBM Security Key Lifecycle Manager server.
port	Specify the port number on which the IBM Security Key Lifecycle Manager server listens for requests.

Request Headers

Header name	Value
Content-Type	application/json
Accept	application/json
Authorization	SKLMAuth userAuthId=<authIdValue>
Accept-Language	Any valid locale that is supported by IBM Security Key Lifecycle Manager. For example: en or de

Request body

JSON object with the following specification:

Property name	Description
type	Required. Specify a value such as certreq to create a certificate generate request.
algorithm	Required. Specify the algorithm with the following values: • RSA • ECDSA
alias	Required. Specify a unique name for the certificate. Retain a record of the alias value of the certificate request, for use when you import the returned certificate.
cn	Required. Specify the common name.
country	Specify a country as a two-letter country code.
fileName	Required. Specify the name of the certificate request file, which is created on the IBM Security Key Lifecycle Manager server, relative to the SKLM_HOME directory. SKLM_HOME is the base directory that contains the IBM Security Key Lifecycle Manager code and configuration.
locality	Specify a locality, such as city.
o	Specify the organization name. For example: o=myCompanyName
ou	Specify the organizational unit name. For example: ou=marketing
state	Specify the full name of a state or province.

Request body

JSON object with the following specification:

Property name	Description
`usage`	Specify the target application usage, such as `SSLSERVER`, with the following values: **3592** Specifies the `3592` device group. **DS8000** Specifies the `DS8000` device group. **GENERIC** Specifies a device family that uses the Key Management Interoperability Protocol to interact with IBM Security Key Lifecycle Manager. The `GENERIC` device group enables management of KMIP objects. Do not use the REST Service interface to add a device to the `GENERIC` device group or to change a `GENERIC` device group attribute. **SSLCLIENT** Specifies the client-side certificate that is used in secure communication by using Secure Socket Layer protocol to authenticate the client device. **SSLSERVER** Specifies the server-side certificate that is used in secure communication by using Secure Socket Layer protocol. **userdevicegroup** Specifies a user-defined group that is based on a supported device family.
`validity`	Required. Specify a time interval in days during which the certificate is valid. The interval can range between 1 day and 9000 days.

Response

Response Headers

Header name	Value and description
`Status Code`	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.
`Content-Type`	`application/json`
`Content-Language`	Locale for the response message.

Success response body

JSON object with the following specification:

JSON property name	Description
code	Returns the code that is specified by the status property.
status	Returns the status to indicate whether the creation of certificate generate request was successful.

Error Response Body

JSON object with the following specification:

JSON property name	Description
code	Returns the application error code.
message	Returns a message that describes the error.

Examples

Service request to create certificate generation request

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

Success response

```
 Status Code: 200 OK
{"code":"0","status":"Succeeded"}
```

Error response

```
Status Code: 400 Bad Request
{"code":"CTGKM0525E","message":"CTGKM0525E Parameter value(s) are not
valid., validity=9999"}
```

Certificate Default Rollover Add REST Service

Use **Certificate Default Rollover Add REST Service** to add a default certificate rollover for a specific date and device group. The rollover certificate takes the place of the previous default certificate.

Operation

```
POST
```

URL `https://<host>:<port>/SKLM/rest/v1/certificates/rollover`

Request

Request Parameters

Parameter	Description
host	Specify the IP address or host name of the IBM Security Key Lifecycle Manager server.
port	Specify the port number on which the IBM Security Key Lifecycle Manager server listens for requests.

Request Headers

Header name	Value
Content-Type	application/json
Accept	application/json
Authorization	SKLMAuth userAuthId=<authIdValue>
Accept-Language	Any valid locale that is supported by IBM Security Key Lifecycle Manager. For example: en or de

Request body

JSON object with the following specification:

JSON property name	Description
alias	Required. Specify a name (not case-sensitive) of the existing certificate.
certDefaultType	Required. Specify whether the certificate is used as the system default or partner certificate, or both. You can include the following values: **1** Certificate is the system default. **2** Certificate is the partner certificate. **3** Certificate is used as both the system default and the partner certificate.
effectiveDate	Required. Specify the rollover date on which the certificate becomes the default system or partner certificate. The value is a current or future date in yyyy-MM-dd format.
usage	Required. Specify the device group. You can include the following values: **3592** Specifies the 3592 device group. **userdevicegroup** Specifies a new, user-defined instance of a supported 3592 device family.

Response

Response Headers

Header name	Value and description
Status Code	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.
Content-Type	application/json

Response Headers

Header name	Value and description
Content-Language	Locale for the response message.

Success response body

JSON object with the following specification:

JSON property name	Description
code	Returns the code that is specified by the **status** property.
status	Returns the status to indicate whether the certificate is marked for rollover.

Error Response Body

JSON object with the following specification:

JSON property name	Description
code	Returns the application error code.
message	Returns a message that describes the error.

Examples

Service request to add a certificate for rollover

```
POST https://localhost:9080/SKLM/rest/v1/certificates/rollover
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
{"usage":"3592","alias":"sklmCertificate","certDefaultType":"1",
"effectiveDate":"2017-05-30"}
```

Success response

```
Status Code: 200 OK
{"code": "0","status": "Succeeded"}
```

Service request to add a certificate for rollover with wrong usage

```
POST https://localhost:9080/SKLM/rest/v1/certificates/rollover
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
{"usage":"LTO","alias":"sklmCertificate","certDefaultType":"1",
"effectiveDate":"2017-05-30"}
```

Error response

```
Status Code: 400 Bad Request
{"code":"CTGKM0830E","message":"Device group is not valid: LTO"}
```

Certificate Default Rollover Delete REST Service

Use **Certificate Default Rollover Delete REST Service** to remove a certificate rollover that is specified in a rollover list.

Note: You cannot use **Certificate Default Rollover Delete REST Service** to delete the certificate default rollovers that are added by using the **tklmCertDefaultRolloverAdd** CLI command.

Operation

```
DELETE
```

URL `https://<host>:<port>/SKLM/rest/v1/certificates/rollover/{uuid}`

Request

Request Parameters

Parameter	Description
host	Specify the IP address or host name of the IBM Security Key Lifecycle Manager server.
port	Specify the port number on which the IBM Security Key Lifecycle Manager server listens for requests.

Request Headers

Header name	Value
Content-Type	`application/json`
Accept	`application/json`
Authorization	`SKLMAuth userAuthId=<authIdValue>`
Accept-Language	Any valid locale that is supported by IBM Security Key Lifecycle Manager. For example: `en` or `de`

Path parameters

JSON property name	Description
uuid	Required. Specify the universal unique identifier of an existing certificate rollover.

Response

Response Headers

Header name	Value and description
Status Code	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.
Content-Type	`application/json`
Content-Language	Locale for the response message.

Success response body

JSON object with the following specification:

JSON property name	Description
status	Returns the status to indicate the removal of certificate rollover.

Error Response Body

JSON object with the following specification:

JSON property name	Description
code	Returns the application error code.
message	Returns a message that describes the error.

Examples

Service request to delete a default certificate rollover with uuid specified

```
DELETE https://localhost:9080/SKLM/rest/v1/certificates/rollover/1234
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
```

Success response

```
Status Code: 200 OK
{"status","Successful"}
```

Service request to delete a default certificate rollover when an incorrect uuid is specified

```
DELETE https://localhost:9080/SKLM/rest/v1/certificates/rollover/101
Content-Type: application/json
Accept : application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
```

Error response

```
Status Code: 500 Internal Server Error
{"code":"CTGKM1009E","message":"CTGKM1009E  Error in scheduler task
detected:Original error message: SCHD0061E: The task information for
task ID 101 and owner token ADMIN was not found in the database."}
```

Certificate Default Rollover List REST Service

Use **Certificate Default Rollover List REST Service** to list certificate rollovers in a rollover list for a specified device group.

Operation
`GET`

URL `https://<host>:<port>/SKLM/rest/v1/certificates/rollover?name=<name value>&usage<usage value>&uuid=<uuid value>`

Request

Request Parameters

Parameter	Description
host	Specify the IP address or host name of the IBM Security Key Lifecycle Manager server.
port	Specify the port number on which the IBM Security Key Lifecycle Manager server listens for requests.

Request Headers

Header name	Value
Content-Type	application/json
Accept	application/json
Authorization	SKLMAuth userAuthId=<authIdValue>

Request Headers

Header name	Value
Accept-Language	Any valid locale that is supported by IBM Security Key Lifecycle Manager. For example: en or de

Query parameters

JSON property name	Description
name	Optional. Specify the name of the existing certificate, which is not case-sensitive.
usage	Required. Specify the device group. You can include the following values: **3592** Specifies the 3592 device group. **userdevicegroup** Specifies a new, user-defined instance of a supported 3592 device family.
uuid	Optional. Specify the unique universal identifier of an existing certificate rollover.

Response

Response Headers

Header name	Value and description
Status Code	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.
Content-Type	application/json
Content-Language	Locale for the response message.

Success response body

JSON array that contains JSON objects with the following specification:

JSON property name	Description
Certificate rollover uuid	Returns the unique universal identifier of the certificate rollover.
<deviceGroup> system default	Returns the system default certificate name/alias for the device group. This response is returned if the certificate is a system default.
<deviceGroup> partner default	Returns the partner default certificate name/alias for the device group. This response is returned if the certificate is a partner default.

Success response body

JSON array that contains JSON objects with the following specification:

JSON property name	Description
Effective date	Returns the rollover date on which the certificate becomes the default system or partner certificate. The value is a current or future date in yyyy-MM-dd format.

Error Response Body

JSON object with the following specification:

JSON property name	Description
code	Returns the application error code.
message	Returns a message that describes the error.

Examples

Service request to list certificate rollover

```
GET https://localhost:9080/SKLM/rest/v1/certificates/rollover?usage=3592
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
```

Success response

```
Status Code: 200 OK
[{"Certificate rollover uuid":"1234","3592 system default":
"3592SysDef",
"Effective date":"2017-05-30"}]
```

Service request to list certificate rollover when an incorrect usage is specified

```
GET https://localhost:9080/SKLM/rest/v1/certificates/rollover?usage=LTT
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
```

Error response

```
Status Code: 400 Bad Request
{"code":"CTGKM0830E","message":"Device group is not valid: LTT"}
```

Certificate Export REST Service

Use **Certificate Export REST Service** to export a certificate file.

Operation

PUT

URL https://*<host>*:*<port>*/SKLM/rest/v1/certificates/export

Request

Request Parameters

Parameter	Description
host	Specify the IP address or host name of the IBM Security Key Lifecycle Manager server.
port	Specify the port number on which the IBM Security Key Lifecycle Manager server listens for requests.

Request Headers

Header name	Value
Content-Type	application/json
Accept	application/json
Authorization	SKLMAuth userAuthId=<authIdValue>
Accept-Language	Any valid locale that is supported by IBM Security Key Lifecycle Manager. For example: en or de

Request body

JSON object with the following specification:

Property name	Description
uuid	Specify the Universal Unique Identifier of the certificate.
fileName	Specify the path and file name to which the certificate is exported. If you specify no path or a relative path, the command appends the file and the path to the *SKLM_HOME* directory if you specify it. If you specify an absolute path, the file is stored in that path; it is *not* relative to the *SKLM_HOME* directory.
format	Optional. Specify any of the following formats for file content: • base64 • DER (Distinguished Encoding Rules)

Response

Response Headers

Header name	Value and description
Status Code	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.
Content-Type	application/json
Content-Language	Locale for the response message.

Success response body

JSON object with the following specification:

JSON property name	Description
code	Returns a 0 (zero) to indicate the completion of the certificate export task.
status	Returns the status with an appropriate message to indicate whether the certificate is exported.

Error Response Body

JSON object with the following specification:

JSON property name	Description
code	Returns the application error code.
message	Returns a message that describes the error.

Examples

Service request to export a certificate

```
PUT https://localhost:9080/SKLM/rest/v1/certificates/export
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
{"uuid":"CERTIFICATE-78d68704-fdde-42df-95da-debef9de930","format":"DER",
"fileName":"/mycertificate.der"}
```

Success response

```
Status Code: 200 OK
{"code":"0","status":"Succeeded"}
```

Error response for an invalid request

```
PUT https://localhost:9080/SKLM/rest/v1/certificates/export
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
{"uuid":"CERTIFICATE-78d68704-fdde-42df-95da-debef9de930","format":"ABC",
"fileName":"/newcertificate.der"}
```

Error response

```
Status Code: 400 Bad Request
{"code":"CTGKM0521E","message":"CTGKM0521E Unsupported certificate
format: ABC"}}
```

Certificate Import REST Service

Use **Certificate Import REST Service** to import a certificate file. You must use the **Certificate Export REST Service** to export the certificates. You can then import this certificate from the exported file.

Operation
`POST`

URL https://*<host>*:*<port>*/SKLM/rest/v1/certificates/import

Request

Request Parameters

Parameter	Description
host	Specify the IP address or host name of the IBM Security Key Lifecycle Manager server.
port	Specify the port number on which the IBM Security Key Lifecycle Manager server listens for requests.

Request Headers

Header name	Value
Content-Type	application/json
Accept	application/json
Authorization	SKLMAuth userAuthId=<authIdValue>
Accept-Language	Any valid locale that is supported by IBM Security Key Lifecycle Manager. For example: en or de

Request body

JSON object with the following specification:

Property name	Description
alias	Specify a unique name for the certificate.
fileName	Specify the file name to import certificate data. The imported file is stored in IBM Security Key Lifecycle Manager in a keystore location relative to the *SKLM_HOME* directory.
format	Specify any of the following formats for file content: • base64 • DER (Distinguished Encoding Rules)

Request body

JSON object with the following specification:

Property name	Description
usage	Specify the target application usage, such as SSLSERVER. You can specify the following values: **3592** Specifies the 3592 device group. **DS8000** Specifies the DS8000 device group. **GENERIC** Specifies a device family that uses the Key Management Interoperability Protocol to interact with IBM Security Key Lifecycle Manager. The GENERIC device group enables management of KMIP objects. Do not use the REST interface to add a device to the GENERIC device group, or to change a GENERIC device group attribute. **SSLCLIENT** Client-side certificate that is used in secure communication by using Secure Socket Layer protocol to authenticate the client device. **SSLSERVER** Server-side certificate that is used in secure communication by using Secure Socket Layer protocol. **SYSLOG** Syslog server-side certificate that is used in secure communication by using Secure Socket Layer protocol to authenticate the syslog server. **userdevicegroup** Specifies a user-defined group that is based on a supported device family.

Response

Response Headers

Header name	Value and description
Status Code	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.
Content-Type	application/json
Content-Language	Locale for the response message.

Success response body

JSON object with the following specification:

JSON property name	Description
code	Returns a 0 (zero) to indicate the completion of the certificate import task
status	Returns the status with an appropriate message to indicate whether the certificate is imported.

Error Response Body

JSON object with the following specification:

JSON property name	Description
code	Returns the application error code.
message	Returns a message that describes the error.

Examples

Service request to export a certificate

```
POST https://localhost:9080/SKLM/rest/v1/certificates/import
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
{"fileName":"/mycertfilenam.base64","alias","newsklmCert","format":"base64",
"usage":"3592"}
```

Success response

```
 Status Code: 200 OK
{"code":"0","status":"Succeeded"}
```

Error response for an invalid request

```
POST https://localhost:9080/SKLM/rest/v1/certificates/import
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
{"fileName":"/mycertfilenam.base64","alias","newsklmCert","format":"ABC",
"usage":"3592"}
```

Error response

```
Status Code: 400 Bad Request
{"code":"CTGKM0521E","message":"CTGKM0521E Unsupported certificate
format: ABC"}
```

Certificate List REST Service

Use **Certificate List REST Service** to return certificate information, which is based on the criteria such as a specific state.

Certificate List REST Service supports pagination. The request parameters, such as offset and count, are used for pagination. For example, to retrieve the first 10 records for the list, set **offset = 1** and **count = 10**. To retrieve the next 10 records, set **offset = 2** and **count = 10**. If you do not specify values for pagination parameters, the first 2000 records are returned.

Operation

```
GET
```

URL

To retrieve all certificates:

```
https://<host>:<port>/SKLM/rest/v1/certificates
```

Note: Returns the first 2000 records.

To retrieve a specific list:

```
https://<host>:<port>/SKLM/rest/v1/certificates?uuid=<uuid>
&alias=<alias>&attributes=<attributes>&usage=<usage>
```

Note: Returns the first 2000 records.

To retrieve a specific list with pagination:

```
https://<host>:<port>/SKLM/rest/v1/certificates?uuid=<uuid>
&alias=<alias>>&attributes=<state value, trusted
value>&usage=<usage>&offset=<offset>&count=<count>
```

Request

Request parameters

Parameter	Description
host	Specify the IP address or host name of the IBM Security Key Lifecycle Manager server.
port	Specify the port number on which the IBM Security Key Lifecycle Manager server listens for requests.
uuid	Specify the unique ID of the certificate. For example: `CERTIFICATE-b4c70958-446d-42c4-ae3b-8c9e0f44c0fa`
alias	Specify the alias name for the certificate.

Request parameters

Parameter	Description
attributes	Specify the attributes to search. Only the state and trusted attributes are supported. You can specify only one state. **state** You can specify the following values: **pending** A certificate request entry is pending the return of a certificate that is approved and certified by a certificate authority. **pre-active** The object exists. It is not yet usable for any cryptographic purpose. An example is a migrated certificate with a future use time stamp. **active** The object is in operational use for protecting and processing data that might use Process Start Date and Protect Stop Date attributes. For example, protecting includes encryption and signature issue. Processing includes decryption and signature verification. **compromised** The security of the object is suspect. A compromised object never returns to an uncompromised state. It cannot be used to protect data. Use the object only to process cryptographically protected information in a client that is trusted to handle compromised cryptographic objects. IBM Security Key Lifecycle Manager retains the state of the object immediately before it was compromised. To process data that was previously protected, the compromised object might continue to be used. **deactivated** Do not use the object to apply cryptographic protection, such as encryption or signing. However, if extraordinary circumstances occur, the object can be used with special permission to process cryptographically protected information. For example, processing includes decryption or verification. **destroyed** Object is no longer usable for any purpose. However, the compromised status of the object can be retained for audit or security purposes. **destroyed-compromised** Object is no longer usable for any purpose. However, the compromised status of the object can be retained for audit or security purposes. **trusted** Values for this attribute can be y, n, or no value. Set the value to y to list only trusted certificates. Set the value to n to list only the untrusted certificates. Not setting a value lists both trusted and untrusted certificates.

Request parameters

Parameter	Description
usage	Specify the target application usage with the following values: **3592** Specifies the 3592 device group. **DS8000** Specifies the DS8000 device group. **GENERIC** Specifies a device family that uses the Key Management Interoperability Protocol to interact with IBM Security Key Lifecycle Manager. The GENERIC device group enables management of KMIP objects Do not use the REST interface to add a device to the GENERIC device group or to change a GENERIC device group attribute **SSLCLIENT** Specifies a client-side certificate that is used in secure communication by using Secure Socket Layer protocol to authenticate the client device. **SSLSERVER** Specifies a server-side certificate that is used in secure communication by using Secure Socket Layer protocol. **SYSLOG** Syslog server-side certificate that is used in secure communication by using Secure Socket Layer protocol to authenticate the syslog server. **userdevicegroup** Specifies a user-defined group that is based on a supported device family.
offset	Specify the page number from which the records are displayed based on the value that you specify for **count**.
count	Specify the number of records to display on the specified page (**offset**).

Request Headers

Header name	Value
Content-Type	application/json
Accept	application/json
Authorization	SKLMAuth userAuthId=<authIdValue>
Accept-Language	Any valid locale that is supported by IBM Security Key Lifecycle Manager. For example: en or de

Response

Response Headers

Header name	Value and description
Status Code	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.
Content-Type	application/json
Content-Language	Locale for the response message.

Success response body

JSON object with the following specification:

JSON property name	Description
uuid	Returns a unique ID for the certificate.
alias	Returns an alias name for the certificate.
information	Returns important information about the certificate.
key store name	Returns the keystore name that contains the certificate.
key store uuid	Returns the keystore unique ID that contains the certificate.
owner	Returns the certificate owner name.
key state	Indicates the certificate status, such as ACTIVE.
issuer name	Returns the distinguished name of the certificate issuer. The property value is from the **Issuer** field of the certificate.
subject name	Returns the certificate subject name. The X.509 certificates contain the subject distinguished name. The property value is from the **Subject** field of the certificate.
activation date	Returns the certificate activation date.
archive date	Returns the certificate archived date.
compromise date	Returns the date on which the certificate is compromised.
creation date	Returns the certificate creation date.
expiration date	Returns the certificate expiration date.
destroy date	Returns the date on which the certificate is destroyed.
trusted	Indicates whether the certificate is trusted.
has private key	Indicates whether the certificate has a private key.
serial number	Returns the certificate serial number.
hash value	Returns the hash value of the certificate.
usage	Returns the target application usage, such as SSLSERVER.
Certificate Type	Returns the certificate type such as X.509 or PGP.

Success response body

JSON object with the following specification:

JSON property name	Description
Certificate Subject Alternate Name	Indicates the alternative name for the certificate owner.
Cryptographic Algorithm	Represents the cryptographic algorithm that is used by the certificate such as RSA, DSA, DES, 3DES, or AES.
Cryptographic Length	Returns the length of the clear-text cryptographic object in bits.
Cryptographic Usage Mask	Represents the cryptographic usage of a key. For example, Encrypt, Decrypt, or Export.
Operation Policy Name	Indicates the operation policy that controls the key management operations on the cryptographic object.
Contact Information	Represents the contact information.
Revocation Reason	Indicates the reason for revoking the managed cryptographic. For example, compromised, expired, or no longer used.
Name	Returns the name to identify and locate the cryptographic object.
Cryptographic Parameters	Returns the cryptographic parameters for cryptographic operations.
Object Group	Returns the group name that contains the cryptographic objects.
Link	Identifies the target managed cryptographic object by its unique identifier. For certificate objects, the parent certificate for a certificate in a certificate chain.
Digest	Contains the digest value of the key or secret data, such as digest of the key material or digest of the certificate value.
Application Specific Information	Stores information specific to the application by using managed cryptographic object.
Custom Attributes	Returns the vendor defined custom attributes.
Last Changed Date	Returns the date and time of the last change to the contents or attributes of the specified managed cryptographic object.
Compromise Occurrence Date	Returns the date and time of when the managed cryptographic object was first believed to be compromised.
Lease Time	Defines a time interval for a managed cryptographic object. After the lease time, the client cannot use the object without obtaining another lease.

Error Response Body

JSON object with the following specification:

JSON property name	Description
code	Returns the application error code.
message	Returns a message that describes the error.

Examples

Service request to list certificate information

```
GET https://<host>:<port>/SKLM/rest/v1/certificates?offset=1&count=10
Content-Type: application/json
Accept : application/json
Authorization : SKLMAuth userAuthId=37ea1939-1374-4db7-84cd-14e399be2d20
Accept-Language : en
```

Success response

```
Status Code : 200 OK
Content-Language: en
[
  {
    "uuid": "CERTIFICATE-c4d35809-b4c2-40af-90db-f291c251f96e",
    "alias": "sslselfcert ",
    "information": null,
    "key store name": "defaultKeyStore ",
    "key store uuid": "KEYSTORE-f1e85369-1e8f-481b-b6ec-10ce59198e89 ",
    "owner": null,
    "key state": "ACTIVE",
    "issuer name": "CN=SSLSelfCert, OU=Security Systems, O=ibm, L=Bangalore,
     ST=Karnataka, C=In",
    "subject name": "CN=SSLSelfCert, OU=Security Systems, O=ibm, L=Bangalore
     , ST=Karnataka, C=In",
    "activation date": "1/8/13 11:13:35 PM India Standard Time",
    "archive date": "null",
    "compromise date": "null",
    "creation date": "1/8/13 11:13:37 PM India Standard Time",
    "expiration date": "1/8/16 11:13:35 PM India Standard Time",
    "destroy date": "null",
    "trusted": "1",
    "has private key": "TRUE ",
    "serial number": "34348697221015",
    "hash value": "0000: fb b5 00 e1 a2 d4 58 45  2d 68 e6 81 c2 43 c4 7b
     ......XE.h...C..0010: 10 75 44 dd bc 2d 5f e4  42 18 a2 27 0f b0 8f 77
     .uD.....B......w",
    "usage": "SSLSERVER",
    "Certificate Type": null,
    "Certificate Subject Alternate Name": null,
    "Cryptographic Algorithm": null,
    "Cryptographic Length": null,
    "Cryptographic Usage Mask": null,
    "Operation Policy Name": null,
    "Contact Information": null,
    "Revocation Reason": null,
    "Name": null,
    "Cryptographic Parameters": null,
    "Object Group": null,
    "Link": null,
    "Digest": null,
    "Application Specific Information": null,
    "Custom Attributes": null,
    "Type": "CERTIFICATE",
    "Last Changed Date": "1/8/13 11:13:37 PM India Standard Time",
    "Compromise Occurence Date": null,
    "Lease Time": null
  }
]
```

Error response

```
Status Code : 400 Bad Request
Content-Language: en
{"code" "CTGKM6002E"
 , "message": "CTGKM6002E Bad Request: Invalid user authentication ID or
invalid request format."
 }
```

Certificate Update REST Service

Use **`Certificate Update REST Service`** to update attributes or usage for a certificate. For example, you might update the state of the certificate to indicate that its use is compromised.

Operation
`PUT`

URL `https://<host>:<port>/SKLM/rest/v1/certificates`

Request

Request Parameters

Parameter	Description
`host`	Specify the IP address or host name of the IBM Security Key Lifecycle Manager server.
`port`	Specify the port number on which the IBM Security Key Lifecycle Manager server listens for requests.

Request Headers

Header name	Value
`Content-Type`	`application/json`
`Accept`	`application/json`
`Authorization`	`SKLMAuth userAuthId=<authIdValue>`
`Accept-Language`	Any valid locale that is supported by IBM Security Key Lifecycle Manager. For example: `en` or `de`

Request body

JSON object with the following specification:

Property name	Description
`uuid`	Specify the universal unique identifier of the certificate.
`attributes`	Optional. Specify one or more of the following attribute-value pairs: **compromised** Specifies whether the use is compromised. The only value is `y` (compromised). You cannot change a compromised key or certificate to an uncompromised state. **information `informationstring`** Specifies more information about the use of an object. **trusted [y|n]** Specifies whether the use is trusted. Set this value to `y` to mark the key or certificate as trusted. Or, set a value of `n` to mark the key or certificate as not trusted. You cannot set compromised or expired keys or certificates to be trusted.

Request body

JSON object with the following specification:

Property name	Description
`usage`	Optional. Specify the target application usage such as `SSLSERVER`. You can specify the following values: **3592** Specifies the 3592 device group. **DS8000** Specifies the DS8000 device group. **GENERIC** Specifies a device family that uses the Key Management Interoperability Protocol to interact with IBM Security Key Lifecycle Manager. The `GENERIC` device group enables management of KMIP objects. Do not use the REST interface to add a device to the `GENERIC` device group, or to change a `GENERIC` device group attribute. **SSLCLIENT** Client-side certificate that is used in secure communication by using Secure Socket Layer protocol to authenticate the client device. **SSLSERVER** Server-side certificate that is used in secure communication by using Secure Socket Layer protocol. **userdevicegroup** Specifies a user-defined group that is based on a supported device family.

Response

Response Headers

Header name	Value and description
`Status Code`	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.
`Content-Type`	`application/json`
`Content-Language`	Locale for the response message.

Success response body

JSON object with the following specification:

JSON property name	Description
code	Returns the code that is specified by the status property.
status	Returns the status message to indicate whether the certificate is updated or not. **0** The status indicates that the certificate update task succeeded. **1** The status indicates that the certificates are not updated.

Error Response Body

JSON object with the following specification:

JSON property name	Description
code	Returns the application error code.
message	Returns a message that describes the error.

Examples

Service request to update a certificate

```
PUT https://localhost:9080/SKLM/rest/v1/certificates
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
{"uuid":"CERTIFICATE-78d68704-fdde-42df-95da-debef9de9309","attributes":
"trusted y"}
```

Success response

```
 Status Code: 200 OK
{"code":"0","status":"CTGKM0508I Updated certificate metadata"}
```

Error response for an invalid request

```
POST https://localhost:9080/SKLM/rest/v1/certificates
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
{"uuid":"CERTIFICATE-78d68704-fdde-42df-95da-debef9de9309","usage":"DS8000"}
"usage":"3592"}
```

Error response

```
Status Code: 500 Internal Server Error
{"code":"CTGKM1129E","message":" CTGKM1129E Target and source device
groups family type does not match."}
```

Counts REST Service

Use **`Counts REST Service`** to get the counts of various cryptographic objects or devices from the IBM Security Key Lifecycle Manager server.

`Counts REST Service` returns counts for the following count types:

- Pending device count
- Low key groups count
- Pending certificates count
- Expiring certificates count

- Pending client certificate count

Operation
`GET`

URL Returns counts for all count types: `https://<host>:<port>/SKLM/rest/v1/ckms/counts`

Returns count for a single count type: `https://<host>:<port>/SKLM/rest/v1/ckms/counts/<countType>`

Returns counts for multiple count types: `https://<host>:<port>/SKLM/rest/v1/ckms/counts?countType=<value>`

Request

Request parameters

Parameter	Description
host	Specify the IP address or host name of the IBM Security Key Lifecycle Manager server.
port	Specify the port number on which the IBM Security Key Lifecycle Manager server listens for requests.
countType	Optional. Specify the count type for which you must obtain the count. Counts for all the count types are returned if you do not specify this parameter. You can specify multiple comma-separated values.

Request Headers

Header name	Value
Content-Type	`application/json`
Accept	`application/json`
Authorization	`SKLMAuth userAuthId=<authIdValue>`
Accept-Language	Any valid locale that is supported by IBM Security Key Lifecycle Manager. For example: `en` or `de`

Response

Response Headers

Header name	Value and description
Status Code	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.
Content-Type	`application/json`

Response Headers

Header name	Value and description
Content-Language	Locale for the response message.

Success response body

JSON Object with the following specification:

JSON property name	Description
PendingDeviceCount	Returns a count of devices that must accept or reject pending requests.
LowKeyKeyGroupsCount	Returns a count of key groups with 10 percent or fewer available keys.
PendingCertificatesCount	Returns a count of certificates that are returned after you sent certificate requests to a certificate authority.
ExpiringCertificatesCount	Returns a count of certificates that are in use, but expires soon.
PendingClientCertCount	Returns a count of pending client certificates.

Error Response Body

JSON object with the following specification:

JSON property name	Description
code	Returns the application error code.
message	Returns a message that describes the error.

Examples

Service request to get counts for all count types

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

Success response

```
Status Code : 200 OK
Content-Language: en
{"PendingDeviceCount":"10",
"LowKeyKeyGroupsCount":"20",
"PendingCertificatesCount":"20",
"ExpiringCertificatesCount":"20",
"PendingClientCertCount":"20"}
```

Service request to get count for a single count type

```
GET https://localhost:9080/SKLM/rest/v1/ckms/counts?countType=PendingDevice
OR
GET https://<host>:<port>/SKLM/rest/v1/ckms/counts/PendingDevice
Content-Type: application/json
Accept : application/json
Authorization : SKLMAuth userAuthId=37ea1939-1374-4db7-84cd-14e399be2d20
Accept-Language : en
```

Success response

```
Status Code : 200 OK
Content-Language: en
{"PendingDeviceCount" : "10"}
```

Service request to get counts for multiple count types

```
GET https://localhost:9080/SKLM/rest/v1/ckms/counts?countType=
LowKeyKeyGroups,
ExpiringCertificates,PendingClientCert
Content-Type: application/json
Accept : application/json
Authorization : SKLMAuth userAuthId=37ea1939-1374-4db7-84cd-14e399be2d20
Accept-Language : en
```

Success response

```
Status Code : 200 OK
Content-Language: en
{"LowKeyKeyGroupsCount":"20",
"ExpiringCertificatesCount":"20",
"PendingClientCertCount":"20"}
```

Error Responses

```
Status Code : 400 Bad Request
Content-Language: en
{"code":"CTGKM6002E","message":"CTGKM6002E Bad Request: Invalid user
authentication ID or invalid request format."}
```

Delete Certificate REST Service

Use **`Delete Certificate REST Service`** to delete a certificate from the IBM Security Key Lifecycle Manager server. The certificate 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 default or partner.
- Scheduled for a future rollover.
- Active `SSLSERVER` or `IKEV2SERVER` certificate.

Operation

`DELETE`

URL `https://<host>:<port>/SKLM/rest/v1/certificates/{alias}`

Request

Request Parameters

Parameter	Description
host	Specify the IP address or host name of the IBM Security Key Lifecycle Manager server.
port	Specify the port number on which the IBM Security Key Lifecycle Manager server listens for requests.

Request Headers

Header name	Value
Content-Type	`application/json`
Accept	`application/json`
Authorization	`SKLMAuth userAuthId=<authIdValue>`
Accept-Language	Any valid locale that is supported by IBM Security Key Lifecycle Manager. For example: `en` or `de`

Path parameter

Parameter	Description
alias	Specify a unique name of the certificate to be deleted.

Response

Response Headers

Header name	Value and description
Status Code	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.
Content-Type	application/json
Content-Language	Locale for the response message.

Success response body

JSON object with the following specification:

JSON property name	Description
status	Returns the status to indicate the certificate deletion.

Error Response Body

JSON object with the following specification:

JSON property name	Description
code	Returns the application error code.
message	Returns a message that describes the error.

Examples

Service request to delete a certificate

```
DELETE https://localhost:9080/SKLM/rest/v1/certificates/sklmCertificate
Content-Type: application/json
Accept : application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
Accept-Language : en
```

Success response

```
Status Code : 200 OK
Content-Language: en
{"code":"0","status":"Succeeded"}
```

Error response

```
Status Code : 400 Bad Request
Content-Language: en
{"code":"CTGKM6002E","message":"CTGKM6002E Bad Request: Invalid user
authentication ID or invalid request format."}
```

```
Status Code : 500 Bad Request
Content-Language: en
{"code":"CTGKM0567E","message":"CTGKM0567E Cannot find the certificate:
sklmcertificate "}
```

Delete Config Property REST Service

Use **Delete Config Property REST Service** to delete one or more properties from the SKLMConfig.properties file, which controls the IBM Security Key Lifecycle Manager server operations.

Operation
DELETE

URL https://*<host>*:*<port>*/SKLM/rest/v1/configProperties

Request

Request Parameters

Parameter	Description
host	Specify the IP address or host name of the IBM Security Key Lifecycle Manager server.
port	Specify the port number on which the IBM Security Key Lifecycle Manager server listens for requests.

Request Headers

Header name	Value
Content-Type	application/json
Accept	application/json
Authorization	SKLMAuth userAuthId=<authIdValue>
Accept-Language	Any valid locale that is supported by IBM Security Key Lifecycle Manager. For example: en or de

Request body

JSON object with the following specification:

JSON property name	Description
propertyName	Specify the configuration property names that you want to delete. You can specify multiple comma-separated properties.

Response

Response Headers

Header name	Value and description
Status Code	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.
Content-Type	`application/json`
Content-Language	Locale for the response message.

Success response body

JSON object with the following specification:

JSON property name	Description
property	Returns the property name that is deleted.
status	Returns the status to indicate the deletion of server property from the configuration file.

Note: The success response code `200 OK` is returned even if the property you requested is not found. An appropriate message is returned in the response body.

Error Response Body

JSON object with the following specification:

JSON property name	Description
code	Returns the application error code.
message	Returns a message that describes the error.

Examples

Service request to delete a single server configuration property

```
DELETE https://localhost:9080/SKLM/rest/v1/configProperties
Content-Type: application/json
Accept : application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
Accept-Language : en
{"propertyName" : "fips"}
```

Success response

```
Status Code : 200 OK
Content-Language: en
[{"property":"fips","status":"CTGKM0606I Update successful,
change will take effect immediately"}]
```

Error response

```
Status Code : 400 Bad Request
Content-Language: en
{"code" : "CTGKM6002E", "message" : "CTGKM6002E Bad Request: Invalid
user authentication ID or invalid request format."}
```

Service request to delete multiple server configuration properties

```
DELETE https://localhost:9080/SKLM/rest/v1/configProperties
Content-Type: application/json
Accept : application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
Accept-Language : en
{"propertyName" : "KMIPListener.ssl.port,FIPS"}
```

Success response

```
Status Code : 200 OK
Content-Language: en
[{"property":"KMIPListener.ssl.port","status":"CTGKM0607I Update
successful, server restart requi-red for change to take effect"},{"
property":"FIPS","status":"CTGKM0556E Cannot find the property in
configuration file: FIPS "}]
```

Error response

```
Status Code : 400 Bad Request
Content-Language: en
{"code" : "CTGKM6002E", "message" : "CTGKM6002E Bad Request: Invalid
user authentication ID or invalid request format."}
```

Delete Key REST Service

Use **Delete Key REST Service** to delete a key entry from the keystore.

Operation

DELETE

URL https://<*host*>:<*port*>/SKLM/rest/v1/keys/<Keyalias>

Request

Request Parameters

Parameter	Description
host	Specify the IP address or host name of the IBM Security Key Lifecycle Manager server.
port	Specify the port number on which the IBM Security Key Lifecycle Manager server listens for requests.

Request Headers

Header name	Value
Content-Type	application/json
Accept	application/json
Authorization	SKLMAuth userAuthId=<authIdValue>
Accept-Language	Any valid locale that is supported by IBM Security Key Lifecycle Manager. For example: en or de

Response

Response Headers

Header name	Value and description
Status Code	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.
Content-Type	`application/json`
Content-Language	Locale for the response message.

Success response body

JSON object with the following specification:

JSON property name	Description
status	Returns the status to indicate the key deletion.

Error Response Body

JSON object with the following specification:

JSON property name	Description
code	Returns the application error code.
message	Returns a message that describes the error.

Examples

Service request to delete a key entry from the keystore

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

Success response

```
Status Code: 200 OK
{"code":"0","status":"Succeeded"}
```

Error response

```
Status Code: 400 Bad Request
{"code" : "CTGKM6002E", "message" : "CTGKM6002E Bad Request: Invalid user
authentication ID or invalid request format."}
```

```
Status Code : 500 Internal Server Error
{"code":"CTGKM0565E","message":"CTGKM0565E Cannot find the key: test "}
```

Delete Replication Config Property REST Service

Use **Delete Replication Config Property REST Service** to delete one or more properties from the ReplicationSKLMConfig.properties configuration file to control the replication operation.

Operation
DELETE

URL https://*<host>*:*<port>*/SKLM/rest/v1/replicationConfigProperties

Request

Request Parameters

Parameter	Description
host	Specify the IP address or host name of the IBM Security Key Lifecycle Manager server.
port	Specify the port number on which the IBM Security Key Lifecycle Manager server listens for requests.

Request Headers

Header name	Value
Content-Type	application/json
Accept	application/json
Authorization	SKLMAuth userAuthId=<authIdValue>
Accept-Language	Any valid locale that is supported by IBM Security Key Lifecycle Manager. For example: en or de

Request body

JSON property name	Description
property	Specify the replication configuration property names that you want to delete. You can specify multiple comma-separated properties.

Response

Response Headers

Header name	Value and description
Status Code	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.
Content-Type	application/json

Response Headers

Header name	Value and description
Content-Language	Locale for the response message.

Success response body

JSON object with the following specification:

JSON property name	Description
property	Returns the property name that is deleted from the replication configuration file.
status	Returns the delete status to indicate whether the configuration property was deleted. The status includes an appropriate message.

Note: The success response code 200 OK is returned even if the property you requested is not found. An appropriate message is returned in the response body.

Error Response Body

JSON object with the following specification:

JSON property name	Description
code	Returns the application error code.
message	Returns a message that describes the error.

Examples

Service request to delete a single configuration property

```
DELETE https://localhost:9080/SKLM/rest/v1/replicationConfigProperties
Content-Type: application/json
Accept : application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
Accept-Language : en
{"property" : "replication.role"}
```

Success response

```
Status Code : 200 OK
Content-Language: en
[{"property":"replication.role","status":"CTGKM0606I Update
successful, change will take effect immediately"}]
```

Error response

```
Status Code : 400 Bad Request
Content-Language: en
{"code" : "CTGKM6002E", "message" : "CTGKM6002E Bad Request: Invalid
user authentication ID or invalid request format."}
```

Service request to delete multiple configuration properties

```
DELETE https://localhost:9080/SKLM/rest/v1/replicationConfigProperties
Content-Type: application/json
Accept : application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
Accept-Language : en
{"property" : "replication.role,backup.ClientPort1"}
```

Success response

```
Status Code : 200 OK
Content-Language: en
[{"property":"replication.role","status":"CTGKM0607I Delete
successful, server restart required for change to take effect"},
"property":"backup.ClientPort1","status":"CTGKM0556E Cannot find the
roperty in configuration file: backup.ClientPort1 "}]
```

Error response

```
Status Code : 400 Bad Request
Content-Language: en
{"code" : "CTGKM6002E", "message" : "CTGKM6002E Bad Request: Invalid
user authentication ID or invalid request format."}
```

Device Add REST Service

Use **`Device Add REST Service`** to add a device to the IBM Security Key Lifecycle Manager database. If the device is a DS5000 storage server, this service can optionally create a system-to-device relationship.

Operation
`POST`

URL `https://<host>:<port>/SKLM/rest/v1/devices`

Request

Request Parameters

Parameter	Description
host	Specify the IP address or host name of the IBM Security Key Lifecycle Manager server.
port	Specify the port number on which the IBM Security Key Lifecycle Manager server listens for requests.

Request Headers

Header name	Value
Content-Type	application/json
Accept	application/json
Authorization	SKLMAuth userAuthId=<authIdValue>
Accept-Language	Any valid locale that is supported by IBM Security Key Lifecycle Manager. For example: en or de

Request body

JSON object with the following specification:

Property name	Description
`serialNumber`	Specify the serial number as an ASCII string. The value is case-sensitive. You can use alphanumeric characters and the special characters such as periods, spaces, dashes, semicolons, and underscore. Do not use a space at the beginning or end of a serial number. **LTO tape drives** The serial number must be exactly 10, 12, or 24 characters in length. IBM Security Key Lifecycle Manager pads a serial number that is 10 characters in length with two leading zeros. **3592 tape drives and DS8000 Turbo drives** The case-sensitive value must be exactly 12 characters in length. **DS5000 storage servers** The serial number can vary 1- 48 characters in length. No padding occurs.
`type`	Specify any of the following device group: **LTO** Specifies the `LTO` device group. **3592** Specifies the 3592 device group. **DS5000** Specifies the DS5000 device group. **DS8000** Specifies the DS8000 device group. **BRCD_ENCRYPTOR** Specifies the `BRCD_ENCRYPTOR` device group that is in the LTO device family. **ONESECURE** Specifies the `ONESECURE` device group that is in the DS5000 device family. **GENERIC** Specifies a device family that uses the Key Management Interoperability Protocol to interact with IBM Security Key Lifecycle Manager. The `GENERIC` device group enables management of KMIP objects. Do not use the REST interface to add a device to the `GENERIC` device group, or to change a `GENERIC` device group attribute. **ETERNUS_DX** Specifies the `ETERNUS_DX` device group that is in the `DS5000` device family. **XIV** Specifies the `XIV` device group that is in the `DS5000` device family. **userdevicegroup** Specifies a user-defined group that is based on a supported device family.

Request body

JSON object with the following specification:

Property name	Description
`attributes`	Optional. Specify one or more attribute-value pairs. Their values are stored in the IBM Security Key Lifecycle Manager database. **aliasOne** Specifies a default alias for a certificate that is used by a 3592 tape drive or a DS8000 Turbo drive. Not used for an `LTO` tape drive or DS5000 storage server. • 3592 tape drive Optional for a 3592 tape drive and specifies the primary certificate that the device uses if the secondary certificate is not available. If this attribute is not specified, the partner default certificate is used, as specified by a table entry for the device group in the IBM Security Key Lifecycle Manager database. • DS8000 Turbo drive Optional for a DS8000 Turbo drive and matches the label “Primary certificate for image” in the graphical user interface panels for a DS8000 Turbo drive.
	aliasTwo Used for a 3592 tape drive or a DS8000 Turbo drive. Not used for an `LTO` tape drive or DS5000 storage server. • 3592 tape drive A default alternative alias for a 3592 tape drive. This value can be the same or different from the value that is specified for the primary certificate. The value specifies the secondary certificate that the device in the 3592 device family uses if the primary certificate is not available. If this attribute is not specified, the partner default certificate is used, as specified by a table entry for the device group in the IBM Security Key Lifecycle Manager database. • DS8000 Turbo drive For a device in the DS8000 device family, the value specifies a secondary certificate that is available for use. For example, you might use this certificate to unlock a DS8000 Turbo drive in the case of a deadlock condition.

Request body

JSON object with the following specification:

Property name	Description
	description Specifies more information that describes the type of device or its purpose. **deviceText** Optional. Specifies the unique text with a minimum length greater than zero bytes and a maximum length of 96 bytes that describes a DS5000 storage server. **driveCert** Specifies the actual certificate that is used to identify the device in `base64` encoded format. For current devices, this field is not in use. **keyPrefix** Specifies a key prefix as part of the key name. To add new keys, specify the prefix and number of keys. This value is used only for a device in the DS5000 device family. **machineID** Optional unless you want to add the association of a device to an existing machine identifier in the IBM Security Key Lifecycle Manager database. Specifies a unique machine ID for a DS5000 storage server, which is a concatenation of the worldwide name and the volume serial number. For example, specify 3042383030303437000000000000. **numberOfKeys** Specifies the number of keys to generate. The keys use the value of the `keyPrefix` attribute. The maximum number of keys is 12. If this value is not specified, the default value is 12 keys. This value is used only for a device in the DS5000 device family. **symAlias** Specifies an alias that is used to identify an existing key group for an `LTO` tape drive. The attribute is also used for DS5000 storage server to change or associate a new device key container. The value of `symAlias` is used to specify which symmetric key group is used to obtain a key for a new device media instance. If this attribute is not specified, then the value of the `symmetricKeySet` attribute is used. For backward compatibility with the Encryption Key Manager product, you can also specify the alias of an existing key entry. **worldwideName** Specifies the name of a device. This name is a nonsecure address that is used in combination with other device information, such as a serial number, to define devices and device paths. Specify a 16-character hexadecimal value that contains only the characters `ABCDEFabcdef1234567890`.

Response

Response Headers

Header name	Value and description
Status Code	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.
Content-Type	`application/json`
Content-Language	Locale for the response message.

Success response body

JSON object with the following specification:

JSON property name	Description
status	Returns the status to indicate whether the device is added with an appropriate message.

Error response body

JSON object with the following specification:

JSON property name	Description
code	Returns the application error code.
message	Returns a message that describes the error.
status	Optional parameter. Returns more information about the error. For example, the status in the following "Error response" example indicates that the **serialNumber** that is added for the device was padded with leading zero's ('00').

Examples

Service request to add a device

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

Success response

```
 Status Code : 200 OK
Content-Language: en
{"code":"0","status":"Succeeded"}
```

Error response when device exists

```
POST https://localhost:9080/SKLM/rest/v1/devices
Content-Type: application/json
Accept : application/json
Authorization : SKLMAuth userAuthId=37ea1939-1374-4db7-84cd-14e399be2d20
Accept-Language : en
{"type":"LTO","serialNumber":"FAA49403AQ"}
```

Error response

```
 Status Code : 500 Internal Server Error
Content-Language: en
{ "code": "CTGKM0648I",
  "message": "CTGKM0648I Device serial number 00FAA49403AQ already
   exists.",
  "status": "CTGKM0254I The device serial number was converted to 12
   characters by adding two leading zeros. Modified device serial
   number " 00FAA49403AQ " ."
}
```

Device Delete REST Service

Use **`Device Delete REST Service`** to remove information that identifies a device from the IBM Security Key Lifecycle Manager database. If the device in the `DS5000` device family and the machine affinity is enabled, deletion of the device removes relationship between a device and a machine.

Operation
`DELETE`

URL `https://<host>:<port>/SKLM/rest/v1/devices/{uuid}`

Request

Request Parameters

Parameter	Description
host	Specify the IP address or host name of the IBM Security Key Lifecycle Manager server.
port	Specify the port number on which the IBM Security Key Lifecycle Manager server listens for requests.

Request Headers

Header name	Value
Content-Type	`application/json`
Accept	`application/json`
Authorization	`SKLMAuth userAuthId=<authIdValue>`
Accept-Language	Any valid locale that is supported by IBM Security Key Lifecycle Manager. For example: `en` or `de`

path parameter

JSON object with the following specification:

Property name	Description
uuid	Required. Specify the universal unique identifier of the device, such as `DEVICE-74386920-148c-47b2-a1e2-d19194b315cf`.

Response

Response Headers

Header name	Value and description
Status Code	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.
Content-Type	`application/json`
Content-Language	Locale for the response message.

Success response body

JSON object with the following specification:

JSON property name	Description
code	Returns the code that is specified by the **status** property.
status	Returns the status to indicate whether the deletion of a device was successful.

Error Response Body

JSON object with the following specification:

JSON property name	Description
code	Returns the application error code.
message	Returns a message that describes the error.

Examples

Service request to delete a device

```
DELETE https://localhost:9080/SKLM/rest/v1/devices/DEVICE-74386920-148c-
47b2-a1e2-d19194b315cf
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
```

Success response

```
Status Code: 200 OK
{"code":"0","status":"Succeeded"}
```

Error response when there is no device

```
DELETE https://localhost:9080/SKLM/rest/v1/devices/DEVICE-74386920-148c
-47b2-a1e2-d19194b315cf
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
```

Error response

```
Status Code: 500 Internal Server Error
{"code":"CTGKM0569E","message":"CTGKM0569E Cannot find the device:
DEVICE-74386920-148c-47b2-a1e2-d19194b315cf"}
```

Device Group Attribute Delete REST Service

Use **Device Group Attribute Delete REST Service** to delete an attribute of a device group, such as myLTO.

Operation

DELETE

URL https://*<host>*:*<port>*/SKLM/rest/v1/deviceGroupAttributes

Request

Request Parameters

Parameter	Description
host	Specify the IP address or host name of the IBM Security Key Lifecycle Manager server.
port	Specify the port number on which the IBM Security Key Lifecycle Manager server listens for requests.

Request Headers

Header name	Value
Content-Type	application/json
Accept	application/json
Authorization	SKLMAuth userAuthId=<authIdValue>
Accept-Language	Any valid locale that is supported by IBM Security Key Lifecycle Manager. For example: en or de

Request body

JSON object with the following specification:

JSON property name	Description
name	Required. Specify a unique device group, such as LTO, with the following values: **LTO** Specifies the LTO device group. **3592** Specifies the 3592 device group. **DS5000** Specifies the DS5000 device group. **DS8000** Specifies the DS8000 device group. **BRCD_ENCRYPTOR** Specifies the BRCD_ENCRYPTOR device group that is in the LTO device family. **ONESECURE** Specifies the ONESECURE device group that is in the DS5000 device family. **GENERIC** Specifies a device family that uses KMIP to interact with IBM Security Key Lifecycle Manager. The GENERIC device group enables management of KMIP objects. Do not use the command-line interface or REST interface to add a device to the GENERIC device group or to change a GENERIC device group attribute. **ETERNUS_DX** Specifies the ETERNUS_DX device group that is in the DS5000 device family. **XIV** Specifies the XIV device group that is in the DS5000 device family. **userdevicegroup** Specifies a user-defined group that is based on a supported device family.

Request body

JSON object with the following specification:

JSON property name	Description
`attribute`	Required. Specify one or more user-defined attribute-value pairs. Use **`Device Group Attribute List REST Service`** to view the current value. You have the following choices: **drive.default.alias1** Specifies the system default certificate that a `3592` device uses if the device is not associated with another certificate. **drive.default.alias2** Specifies the system partner certificate that a `3592` device uses if the device is not associated with another certificate. **symmetricKeySet** Specifies a key group to be used for a device group. **shortName** Specifies a short label that is usually a drive type, such as `LTO`. This label is used for any additional attributes that are required by an original equipment manufacturer. **longName** Specifies an extended descriptive name of a drive type, such as `my division LTO`. For example, this information might include business information.

Response

Response Headers

Header name	Value and description
`Status Code`	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.
`Content-Type`	`application/json`
`Content-Language`	Locale for the response message.

Success response body

JSON object with the following specification:

JSON property name	Description
code	Returns the code that is specified by the **status** property.
status	Returns the status to indicate whether the device group attribute deletion is successful.

Error Response Body

JSON object with the following specification:

JSON property name	Description
code	Returns the application error code.
message	Returns a message that describes the error.

Examples

Service request to delete an attribute

```
DELETE http://localhost:9080/SKLM/rest/v1/deviceGroupAttributes
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
{"name":"myLTO","attributes":"symmetricKeySet"}
```

Success response

```
Status Code: 200 OK
{"code": "0","status": "Succeeded"}
```

Service request to delete an attribute when an invalid parameter is specified

```
DELETE http://localhost:9080/SKLM/rest/v1/deviceGroupAttributes
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
{"Name":"myLTO","attribute":"longName, shortName"}
```

Error response

```
Status Code: 400 Bad Request
{"code":"CTGKM0630E","message":"CTGKM0630E Validation error:
\"Invalid name \" for parameter \"Name\"."}
```

Device Group Attribute List REST Service

Use **Device Group Attribute List REST Service** to list all of the attributes of a device group, such as LTO.

Operation
GET

URL https://*<host>*:*<port>*/SKLM/rest/v1/deviceGroupAttributes?name=<deviceGroupName>

Request

Request Parameters

Parameter	Description
host	Specify the IP address or host name of the IBM Security Key Lifecycle Manager server.

Request Parameters

Parameter	Description
port	Specify the port number on which the IBM Security Key Lifecycle Manager server listens for requests.

Request Headers

Header name	Value
Content-Type	application/json
Accept	application/json
Authorization	SKLMAuth userAuthId=<authIdValue>
Accept-Language	Any valid locale that is supported by IBM Security Key Lifecycle Manager. For example: en or de

Query parameter

JSON object with the following specification:

JSON property name	Description
name	Required. Specify a unique device group, such as LTO, with the following values: **LTO** Specifies the LTO device group. **3592** Specifies the 3592 device group. **DS5000** Specifies the DS5000 device group. **DS8000** Specifies the DS8000 device group. **BRCD_ENCRYPTOR** Specifies the BRCD_ENCRYPTOR device group that is in the LTO device family. **ONESECURE** Specifies the ONESECURE device group that is in the DS5000 device family. **GENERIC** Specifies a device family that uses KMIP to interact with IBM Security Key Lifecycle Manager. The GENERIC device group enables management of KMIP objects. Do not use the command-line interface or REST interface to add a device to the GENERIC device group or to change a GENERIC device group attribute. **ETERNUS_DX** Specifies the ETERNUS_DX device group that is in the DS5000 device family. **XIV** Specifies the XIV device group that is in the DS5000 device family. **userdevicegroup** Specifies a user-defined group that is based on a supported device family.

Response

Response Headers

Header name	Value and description
`Status Code`	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.
`Content-Type`	`application/json`
`Content-Language`	Locale for the response message.

Success response body

JSON array that contains JSON objects with the following specification:

JSON property name	Description
`longName`	Returns the long name of the device group.
`shortName`	Returns the short name of the device group.
`enableKMIPDelete`	Indicates whether deletion is enabled on the object or not. Valid values are `true` or `false`.
`symmetricKeySet`	Returns a key group to be used for `LTO` tape drives.
`drive.default.alias1`	Returns the system default certificate that the device uses if the device is not associated with another certificate.
`drive.default.alias1`	Returns the system partner certificate that the device uses if the device is not associated with another partner certificate.
`device.AutoPendingAutoDiscovery`	Indicates whether to add a device that contacts IBM Security Key Lifecycle Manager to a list of pending devices that you can accept or reject before key serving occurs, or to add a device automatically to the drive table for immediate key service upon request. The attribute applies only to predefined base device families, not to user-defined device groups. You have the following choices: **0 (manual)** Both the auto pending and auto discovery functions are off. **1(auto accept)** The auto discovery function is on, and the auto pending function is off. **2 (auto pending)** The auto pending function is on.

Success response body

JSON array that contains JSON objects with the following specification:

JSON property name	Description
device.enableMachineAffinity	Indicates that the device group is enabled to store the association of a device to an existing machine identifier in the IBM Security Key Lifecycle Manager database. To modify this value, you must have a role with permissions to create or modify and also have permission to the DS5000 device group. Valid values are `true` or `false`.

Error Response Body

JSON object with the following specification:

JSON property name	Description
code	Returns the application error code.
message	Returns a message that describes the error.

Examples

Service request to obtain attributes list of a device group

```
GET https://localhost:9080/SKLM/rest/v1/deviceGroupAttributes?name=3592
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
```

Success response

```
Status Code: 200 OK
[
    {
       "longName": null,
       "shortName": null,
       "enableKMIPDelete": "false",
       "drive.default.alias1": null,
       "drive.default.alias2": null,
       "device.AutoPendingAutoDiscovery": "0"
    }
 ]
```

Service request for attributes list when an invalid device group is specified

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

Error response

```
Status Code: 400 Bad Request
{"code":"CTGKM0830E","message":"Device group is not valid: LT"}
```

Device Group Attribute Update REST Service

Use **Device Group Attribute Update REST Service** to update the attributes of a device group, such as `myLTO`.

Operation

```
PUT
```

URL `https://<host>:<port>/SKLM/rest/v1/deviceGroupAttributes`

Request

Request Parameters

Parameter	Description
host	Specify the IP address or host name of the IBM Security Key Lifecycle Manager server.
port	Specify the port number on which the IBM Security Key Lifecycle Manager server listens for requests.

Request Headers

Header name	Value
Content-Type	application/json
Accept	application/json
Authorization	SKLMAuth userAuthId=<authIdValue>
Accept-Language	Any valid locale that is supported by IBM Security Key Lifecycle Manager. For example: en or de

Request body

JSON object with the following specification:

JSON property name	Description
name	Required. Specify a unique device group, such as LTO, with the following values: **LTO** Specifies the LTO device group. **3592** Specifies the 3592 device group. **DS5000** Specifies the DS5000 device group. **DS8000** Specifies the DS8000 device group. **BRCD_ENCRYPTOR** Specifies the BRCD_ENCRYPTOR device group that is in the LTO device family. **ONESECURE** Specifies the ONESECURE device group that is in the DS5000 device family. **GENERIC** Specifies a device family that uses KMIP to interact with IBM Security Key Lifecycle Manager. The GENERIC device group enables management of KMIP objects. Do not use the command-line interface or REST interface to add a device to the GENERIC device group or to change a GENERIC device group attribute. **ETERNUS_DX** Specifies the ETERNUS_DX device group that is in the DS5000 device family. **XIV** Specifies the XIV device group that is in the DS5000 device family. **userdevicegroup** Specifies a user-defined group that is based on a supported device family.

Request body

JSON object with the following specification:

JSON property name	Description
`attributes`	Required. Specify one or more user-defined attribute-value pairs. Use **`Device Group Attribute List REST Service`** to view the current value. You have the following choices: **drive.default.alias1** Specifies the system default certificate that a `3592` device uses if the device is not associated with another certificate. **drive.default.alias2** Specifies the system partner certificate that a `3592` device uses if the device is not associated with another certificate. **enableKMIPDelete** Enables or disables KMIP delete requests. The **`klmAdminDeviceGroup`** permission permits administration, such as (create, view, delete) of a device group. Disabling this attribute when you create a device group prevents KMIP clients from deleting keys in the device group. The default is disabled (false). Use **`Device Group Attribute Update REST Service`** to modify this attribute. **symmetricKeySet** Specifies a key group to be used for a device group. **shortName** Specifies a short label that is usually a drive type, such as `LTO`. This label is used for any additional attributes that are required by an original equipment manufacturer. **longName** Specifies an extended descriptive name of a drive type, such as `my division LTO`. For example, this information might include business information. **device.enableMachineAffinity** Specifies whether a specific device group is enabled to store the association of a device to an existing machine identifier in the Security Key Lifecycle Manager database. Valid values are `true` or `false`. **device.AutoPendingAutoDiscovery** Adds a device that contacts Security Key Lifecycle Manager to a list of pending devices that you can accept or reject before key serving occurs. Or, adds a device automatically to the drive table for immediate key service upon request. The attribute applies only to predefined base device families, and not to user-defined device groups. You have the following choices: **0 (manual)** Both the auto pending and auto discovery functions are off. **1(auto accept)** The auto discovery function is on, and the auto pending function is off. **2 (auto pending)** The auto pending function is on.

Response

Response Headers

Header name	Value and description
Status Code	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.
Content-Type	application/json
Content-Language	Locale for the response message.

Success response body

JSON object with the following specification:

JSON property name	Description
code	Returns the code that is specified by the **status** property.
status	Returns the status to indicate whether the device group attribute update task is successful.

Error Response Body

JSON object with the following specification:

JSON property name	Description
code	Returns the application error code.
message	Returns a message that describes the error.

Examples

Service request to update an attribute

```
PUT http://localhost:9080/SKLM/rest/v1/deviceGroupAttributes
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
{"name":"3592","attributes":"longName 3592"}
```

Success response

```
Status Code: 200 OK
{"code": "0","status": "Succeeded"}
```

Service request to update an attribute when an invalid parameter is specified

```
PUT http://localhost:9080/SKLM/rest/v1/deviceGroupAttributes
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
{"Name":"3592","attributes":"longName 3592 device, shortName 3592"}
```

Error response

```
Status Code: 400 Bad Request
{"code":"CTGKM0630E","message":"CTGKM0630E Validation error: \
"Invalid name \" for parameter \"Name\"."}
```

Device Group Base List REST Service

Use Device Group Base List REST Service to list all of the device group families that IBM Security Key Lifecycle Manager provides, such as LTO and 3592.

Operation
GET

URL https://*<host>*:*<port>*/SKLM/rest/v1/deviceGroups/base

Request

Request Parameters

Parameter	Description
host	Specify the IP address or host name of the IBM Security Key Lifecycle Manager server.
port	Specify the port number on which the IBM Security Key Lifecycle Manager server listens for requests.

Request Headers

Header name	Value
Content-Type	application/json
Accept	application/json
Authorization	SKLMAuth userAuthId=<authIdValue>
Accept-Language	Any valid locale that is supported by IBM Security Key Lifecycle Manager. For example: en or de

Response

Response Headers

Header name	Value and description
Status Code	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.
Content-Type	application/json
Content-Language	Locale for the response message.

Success response body

JSON object with the following specification:

JSON property name	Description
Device Family	Returns a list of base device families that is supported by IBM Security Key Lifecycle Manager, such as LTO and 3592.

Error Response Body

JSON object with the following specification:

JSON property name	Description
code	Returns the application error code.
message	Returns a message that describes the error.

Examples

Service request to get device group base list

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

Success response

```
Status Code: 200 OK
[
  {  "Device Family": "3592" },
   {  "Device Family": "LTO"  },
   { "Device Family": "DS8000" },
   { "Device Family": "DS5000" },
   { "Device Family": "GENERIC" }
]
```

Error response

```
Status Code: 400 Bad Request
Content-Language: en
{"code":"CTGKM6002E","message":"CTGKM6002E Bad Request: Invalid user
authentication ID or invalid request format."}
```

Device Group Create REST Service

Use **Device Group Create REST Service** to create a device group, such as myLTO. The new device group is a child of a parent device family, such as LTO.

Operation
POST

URL https://*<host>*:*<port>*/SKLM/rest/v1/deviceGroups/{groupName}

Request

Request Parameters

Parameter	Description
host	Specify the IP address or host name of the IBM Security Key Lifecycle Manager server.
port	Specify the port number on which the IBM Security Key Lifecycle Manager server listens for requests.

Request Headers

Header name	Value
Content-Type	application/json
Accept	application/json
Authorization	SKLMAuth userAuthId=<authIdValue>
Accept-Language	Any valid locale that is supported by IBM Security Key Lifecycle Manager. For example: en or de

Path parameters

Parameter name	Description
groupName	Specify a user-defined name for a device group. For example, myDivisionLTO Follow these rules to define a name: • Do not specify a reserved value of 3592, DS8K, DS8000, LTO, DS5000, or GENERIC. • Do not specify a reserved value of SSLSERVER or SSLCLIENT. • The name must start with an alphabetic character, not a numeral. It can contain only alphanumeric characters and underscores. • The name cannot consist of a single underscore and must not exceed a length of 16 characters.

Request body

JSON object with the following specification:

JSON property name	Description
deviceFamily	Specify an existing device family that IBM Security Key Lifecycle Manager provides. You can specify the following device family: **LTO** Specifies the LTO device family. **3592** Specifies the 3592 device family. **DS5000** Specifies the DS5000 device family. **GENERIC** Specifies a device family that uses KMIP to interact with IBM Security Key Lifecycle Manager. The GENERIC device group enables management of KMIP objects.
device.enableMachineAffinity	Specify the device groups in the DS5000 device family that enabled to store the association of a device to an existing system identifier in the IBM Security Key Lifecycle Manager database. The values are `true` (enable) or `false` (disable). An instance of the property is stored for each device group.
enableKMIPDelete	Enables or disables KMIP delete requests. Disabling this attribute when you create a device group prevents KMIP clients from deleting keys in the device group. Default is false (disabled).
shortName	Specify a short label that is usually a drive type such as LTO. This property is used for any additional attributes that are needed by an original equipment manufacturer.
longName	Specify an extended descriptive name of a drive type, such as `my division LTO`. For example, it might include business information.

Response

Response Headers

Header name	Value and description
Status Code	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.
Content-Type	`application/json`
Content-Language	Locale for the response message.

Success response body

JSON object with the following specification:

JSON property name	Description
DeviceGroupName	Returns the name of the group that is created.

Error Response Body

JSON object with the following specification:

JSON property name	Description
code	Returns the application error code.
message	Returns a message that describes the error.

Examples

Service request to create a device group

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

Success response

```
Status Code: 200 OK
{"code":"0","status":"Succeeded"}
```

Error response

```
Status Code: 400 Bad Request
{"code":"CTGKM6002E","message":"CTGKM6002E Bad Request: Invalid user
authentication ID or invalid request format."}
```

Device Group Delete REST Service

Use **Device Group Delete REST Service** to delete an empty customized device group such as myLTO. You cannot delete a device group if it contains any devices, keys, or certificates. You also cannot delete a device family that IBM Security Key Lifecycle Manager provides.

Operation

DELETE

URL https://*<host>*:*<port>*/SKLM/rest/v1/deviceGroups/{name}

Request

Request Parameters

Parameter	Description
host	Specify the IP address or host name of the IBM Security Key Lifecycle Manager server.
port	Specify the port number on which the IBM Security Key Lifecycle Manager server listens for requests.

Request Headers

Header name	Value
Content-Type	application/json
Accept	application/json
Authorization	SKLMAuth userAuthId=<authIdValue>
Accept-Language	Any valid locale that is supported by IBM Security Key Lifecycle Manager. For example: en or de

Path parameters

Parameter name	Description
name	Specify a user-defined name of an existing device group. You cannot delete a device family that a customized device group references.

Response

Response Headers

Header name	Value and description
Status Code	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.
Content-Type	application/json
Content-Language	Locale for the response message.

Success response body

JSON object with the following specification:

JSON property name	Description
status	Returns the status to indicate deletion of the specified device group.

Error Response Body

JSON object with the following specification:

JSON property name	Description
code	Returns the application error code.
message	Returns a message that describes the error.

Examples

Service request to delete an empty device group

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

Success response

```
 Status Code: 200 OK
{"code":"0","status":"Succeeded"}
```

Service request to delete a device group that is not empty

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

Error response

```
 Status Code: 500 Internal Server Error
{"code":"CTGKM1130E","message":"CTGKM1130E Cannot delete a device
group when keys, certificates, groups, or devices are attached to
that device group."}
```

Device Group List REST Service

Use **`Device Group List REST Service`** to obtain a list of device groups within a device family, such as `LTO`.

`Device Group List REST Service` supports pagination. The request parameters, such as `offset` and `count`, are used for pagination. For example, to retrieve the first 10 records for the list, set **`offset = 1`** and **`count = 10`**. To retrieve the next 10 records, set **`offset = 2`** and **`count = 10`**. If you do not specify values for pagination parameters, the first 2000 records are returned.

Operation

`GET`

URL

To retrieve all device groups:

`https://<host>:<port>/SKLM/rest/v1/deviceGroups`

Note: Returns the first 2000 records.

To retrieve a specific list with pagination:

`https://<host>:<port>/SKLM/rest/v1/deviceGroups?offset=<offset>&count=<count>`

Request

Request Parameters

Parameter	Description
host	Specify the IP address or host name of the IBM Security Key Lifecycle Manager server.
port	Specify the port number on which the IBM Security Key Lifecycle Manager server listens for requests.

Request Headers

Header name	Value
Content-Type	application/json

Request Headers

Header name	Value
Accept	application/json
Authorization	SKLMAuth userAuthId=<authIdValue>
Accept-Language	Any valid locale that is supported by IBM Security Key Lifecycle Manager. For example: en or de

Query parameters

Parameter name	Description
name	Specify a user-defined name for a device group. For example, BRCD_ENCRYPTOR.
deviceFamily	Specify a unique device family, such as LTO. You can specify the following values: **LTO** Specifies the LTO device family. **3592** Specifies the 3592 device family. **DS5000** Specifies the DS5000 device family. **GENERIC** Specifies a device family that uses KMIP to interact with IBM Security Key Lifecycle Manager. The GENERIC device group enables management of KMIP objects.
offset	Specify the page number from which the records are displayed based on the value that you specify for **count**.
count	Specify the number of records to display on the page that you specified with **offset**. The count must not exceed 2000 records.

Response

Response Headers

Header name	Value and description
Status Code	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.
Content-Type	application/json
Content-Language	Locale for the response message.

Success response body

JSON object with the following specification:

JSON property name	Description
Device Group UUID	Returns a unique identifier of the device group.
Device Group Name	Returns the user-defined name of the device group. For example, BRCD_ENCRYPTOR.
symmetricKeySet	Returns the key group for a device type.
drive.default.alias1	Returns the default alias for the device type. Value can be the same or different from **drive.default.alias2**.
drive.default.alias2	Returns the default alias for the device type. Value can be the same or different from **drive.default.alias1**.
shortName	Returns a short label that is usually a drive type, such as LTO. You can use this property for any additional attributes that are needed by an original equipment manufacturer.
longName	Returns an extended descriptive name of the device type, such as **my division LTO**. For example, this information might include business information.
roleName	Indicates the user role name.
device.AutoPendingAutoDiscovery	Indicates what to do with a new device that contacts IBM Security Key Lifecycle Manager. You have the following choices: **0 (manual)** All incoming devices are rejected, and not added to the data store. You must manually add devices and system IDs. **1 (auto accept)** All incoming devices of a valid device group are added to the data store. They are automatically served keys upon request. **2 (auto pending)** All incoming devices are added to a pending list, but they 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.
Device Family	Returns the device family name, such as LTO.
enableKMIPDelete	Indicates whether the KMIP delete requests are enabled or disabled. Disabling this attribute when you create a device group prevents KMIP clients from deleting keys in the device group. The default is false (disabled).

Error Response Body

JSON object with the following specification:

JSON property name	Description
code	Returns the application error code.
message	Returns a message that describes the error.

Examples

Service request to get a device group list

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

Success response

```
Status Code : 200 OK
Content-Language: en
    [
       {
           "Device Group UUID": "1",
           "Device Group Name": "3592",
           "Device Family": "3592",
           "symmetricKeySet": null,
           "drive.default.alias1": null,
           "drive.default.alias2": null,
           "shortName": null,
           "longName": null,
           "roleName": "TS3592",
           "device.AutoPendingAutoDiscovery": "0",
           "enableKMIPDelete": "false"
       },
       {
           "Device Group UUID": "2",
           "Device Group Name": "LTO",
           "Device Family": "LTO",
           "symmetricKeySet": "keygroup1",
           "drive.default.alias1": null,
           "drive.default.alias2": null,
           "shortName": null,
           "longName": null,
           "roleName": "LTO",
           "device.AutoPendingAutoDiscovery": "0",
           "enableKMIPDelete": "false"
       },
       {
           "Device Group UUID": "3",
           "Device Group Name": "DS8000",
           "Device Family": "DS8000",
           "symmetricKeySet": null,
           "drive.default.alias1": null,
           "drive.default.alias2": null,
           "shortName": null,
           "longName": null,
           "roleName": "DS8000",
           "device.AutoPendingAutoDiscovery": "0",
           "enableKMIPDelete": "false"
       },
       {
           "Device Group UUID": "4",
           "Device Group Name": "DS5000",
           "Device Family": "DS5000",
           "symmetricKeySet": null,
           "drive.default.alias1": null,
           "drive.default.alias2": null,
           "shortName": null,
           "longName": null,
           "roleName": "DS5000",
           "device.AutoPendingAutoDiscovery": "0",
           "enableKMIPDelete": "false"
       },
       {
           "Device Group UUID": "5",
    "Device Group Name": "GENERIC",
    "Device Family": "GENERIC",
    "symmetricKeySet": null,
    "drive.default.alias1": null,
    "drive.default.alias2": null,
    "shortName": null,
    "longName": null,
    "roleName": "GENERIC",
    "device.AutoPendingAutoDiscovery": "0",
    "enableKMIPDelete": "false"
},
{
    "Device Group UUID": "1000",
    "Device Group Name": "BRCD_ENCRYPTOR",
    "Device Family": "LTO",
    "symmetricKeySet": null,
    "drive.default.alias1": null,
    "drive.default.alias2": null,
    "shortName": null,
    "longName": null,
    "roleName": "BRCD_ENCRYPTOR",
    "device.AutoPendingAutoDiscovery": "0",
    "enableKMIPDelete": "false"
},
{
    "Device Group UUID": "1001",
    "Device Group Name": "ONESECURE",
    "Device Family": "DS5000",
    "symmetricKeySet": null,
    "drive.default.alias1": null,
    "drive.default.alias2": null,
    "shortName": null,
    "longName": null,
    "roleName": "ONESECURE",
    "device.AutoPendingAutoDiscovery": "0",
    "enableKMIPDelete": "false"
},
{
    "Device Group UUID": "10000",
    "Device Group Name": "MYLTO",
    "Device Family": "LTO",
    "symmetricKeySet": null,
    "drive.default.alias1": null,
    "drive.default.alias2": null,
    "shortName": "myLTO",
    "longName": "my companyname LTO devices",
    "roleName": "MYLTO",
    "device.AutoPendingAutoDiscovery": "0",
    "enableKMIPDelete": "false"
},
{
    "Device Group UUID": "10001",
    "Device Group Name": "AAA_JAG",
    "Device Family": "3592",
    "symmetricKeySet": null,
    "drive.default.alias1": null,
    "drive.default.alias2": null,
    "shortName": null,
    "longName": null,
    "roleName": "AAA_JAG",
    "device.AutoPendingAutoDiscovery": "0",
    "enableKMIPDelete": "false"
},
{
    "Device Group UUID": "10002",
    "Device Group Name": "BBB_JAG",
    "Device Family": "3592",
       "symmetricKeySet": null,
       "drive.default.alias1": null,
       "drive.default.alias2": null,
       "shortName": null,
       "longName": null,
       "roleName": "BBB_JAG",
       "device.AutoPendingAutoDiscovery": "0",
       "enableKMIPDelete": "false"
   },
   {
       "Device Group UUID": "10003",
       "Device Group Name": "CCC_JAG",
       "Device Family": "3592",
       "symmetricKeySet": null,
       "drive.default.alias1": null,
       "drive.default.alias2": null,
       "shortName": null,
       "longName": null,
       "roleName": "CCC_JAG",
       "device.AutoPendingAutoDiscovery": "0",
       "enableKMIPDelete": "false"
   },
   {
       "Device Group UUID": "10004",
       "Device Group Name": "MYDEV",
       "Device Family": "LTO",
       "symmetricKeySet": null,
       "drive.default.alias1": null,
       "drive.default.alias2": null,
       "shortName": null,
       "longName": null,
       "roleName": "MYDEV",
       "device.AutoPendingAutoDiscovery": "0",
       "enableKMIPDelete": "false"
   }
]
```

Device group list with pagination

```
GET https://localhost:9080/SKLM/rest/v1/deviceGroups?offset=1&count=1&name=
myLTO
Content-Type: application/json
Accept : application/json Authorization: SKLMAuth userAuthId=139aeh34567m
Accept-Language : en
```

Success response

```
Status Code : 200 OK
Content-Language: en


    [
       {
           "Device Group UUID": "10000",
           "Device Group Name": "MYLTO",
           "Device Family": "LTO",
           "symmetricKeySet": null,
           "drive.default.alias1": null,
           "drive.default.alias2": null,
           "shortName": "myLTO",
           "longName": "my companyname LTO devices",
           "roleName": "MYLTO",
           "device.AutoPendingAutoDiscovery": "0",
           "enableKMIPDelete": "false"
       }
    ]
```

Error response

```
Status Code : 400 Bad Request
Content-Language: en
{"code":"CTGKM6002E","message":"CTGKM6002E Bad Request: Invalid user
authentication ID or invalid request format."}
```

Device List REST Service

Use **`Device List REST Service`** to list information about all devices of a specific device group, or a device in the IBM Security Key Lifecycle Manager database.

`Device List REST Service` supports pagination. The request parameters, such as `offset` and `count`, are used for pagination. For example, to retrieve the first 10 records for the list, set **`offset = 1`** and **`count = 10`**. To retrieve the next 10 records, set **`offset = 2`** and **`count = 10`**. If you do not specify values for pagination parameters, the first 2000 records are returned.

Operation

`GET`

URL

To retrieve all devices:

`https://<host>:<port>/SKLM/rest/v1/devices`

Note: Returns the first 2000 records.

To retrieve a specific list:

```
https://<host>:<port>/SKLM/rest/v1/devices?type=<type>
&uuid=<uuid>
```

Note: Returns the first 2000 records.

To retrieve a specific list with pagination:

```
https://<host>:<port>/SKLM/rest/v1/devices?type=<type>
&uuid=<uuid>&offset=<offset>&count=<count>
```

Request

Request parameters

Parameter	Description
`host`	Specify the IP address or host name of the IBM Security Key Lifecycle Manager server.
`port`	Specify the port number on which the IBM Security Key Lifecycle Manager server listens for requests.

Request parameters

Parameter	Description
Type	Specify the device group type. The default is all device groups. **LTO** Specifies the LTO device group. **3592** Specifies the 3592 device group. **DS5000** Specifies the DS5000 device group. **DS8000** Specifies the DS8000 device group. **BRCD_ENCRYPTOR** Specifies the BRCD_ENCRYPTOR device group that is in the LTO device family. **ONESECURE** Specifies the ONESECURE device group that is in the DS5000 device family. **GENERIC** Specifies a device family that uses the Key Management Interoperability Protocol to interact with IBM Security Key Lifecycle Manager. The GENERIC device group enables management of KMIP objects. Do not use the REST interface to add a device to the GENERIC device group, or to change a GENERIC device group attribute. **ETERNUS_DX** Specifies the ETERNUS_DX device group that is in the DS5000 device family. **XIV** Specifies the XIV device group that is in the DS5000 device family. *userdevicegroup* Specifies a user-defined group that is based on a supported device family.
uuid	Specify the unique ID of the device. For example: DEVICE-74386920-148c-47b2-a1e2-d19194b315cf
offset	Specify the page number from which the records are displayed based on the value that you specify for **count**.
count	Specify the number of records to display on the specified page (**offset**). The first 2000 records are returned if you do not specify the value for **offset** and **count**.

Request Headers

Header name	Value
Content-Type	application/json
Accept	application/json
Authorization	SKLMAuth userAuthId=<authIdValue>
Accept-Language	Any valid locale that is supported by IBM Security Key Lifecycle Manager. For example: en or de

Response

Response Headers

Header name	Value and description
`Status Code`	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.
`Content-Type`	`application/json`
`Content-Language`	Locale for the response message.

Success response body

JSON object with the following specification:

JSON property name	Description
`Description`	Describes the type of device or its purpose.
`Serial Number`	Returns the serial number as an ASCII string.
`Device uuid`	Returns the unique ID of the device.

Success response body

JSON object with the following specification:

JSON property name	Description
Device group	Returns the device group type. This property can contain the following device groups: **LTO** Specifies the LTO device group. **3592** Specifies the 3592 device group. **DS5000** Specifies the DS5000 device group. **DS8000** Specifies the DS8000 device group. **BRCD_ENCRYPTOR** Specifies the BRCD_ENCRYPTOR device group that is in the LTO device family. **ONESECURE** Specifies the ONESECURE device group that is in the DS5000 device family. **GENERIC** Specifies a device family that uses KMIP to interact with IBM Security Key Lifecycle Manager. The GENERIC device group enables management of KMIP objects. Do not use the command-line interface or REST interface to add a device to the GENERIC device group or to change a GENERIC device group attribute. **ETERNUS_DX** Specifies the ETERNUS_DX device group that is in the DS5000 device family. **XIV** Specifies the XIV device group that is in the DS5000 device family. **userdevicegroup** Specifies a user-defined group that is based on a supported device family.
World wide name	Returns the device name, which is a nonsecure address, which is used in combination with other device information, such as a serial number. You can use this name to define devices and device paths.
Sym alias	Returns an alias to identify an existing key group for an LTO tape drive. The attribute is also used for the DS5000 storage server to change or associate a new device key container.
Host address	Returns the host address for the device.
Key alias 1	Returns the default key alias for a certificate that is used by a 3592 tape drive or a DS8000 Turbo drive. This alias is not used for an LTO tape drive or DS5000 storage server.
Key alias 2	Returns a key alias for a 3592 tape drive or a DS8000 Turbo drive. This alias is not used for an LTO tape drive or DS5000 storage server.
Certificate length	Returns the length of the actual certificate that identifies the device.

Success response body

JSON object with the following specification:

JSON property name	Description
Device Text	Returns a unique text that describes a DS5000 storage server. The text ranges is up to 96 bytes, but it must be greater than 0.
Current Key	Returns the current key for the device.

Error Response Body

JSON object with the following specification:

JSON property name	Description
code	Returns the application error code.
message	Returns a message that describes the error.

Examples

Service request to list device information

```
GET https://<host>:<port>/SKLM/rest/v1/devices
Content-Type: application/json
Accept : application/json
Authorization : SKLMAuth userAuthId=37ea1939-1374-4db7-84cd-14e399be2d20
Accept-Language : en
```

Success response

```
Status Code : 200 OK
Content-Language: en
[
  {
    "Description": "salesDivisionDrive",
    "Serial Number": "FAA49403AQJF",
    "Device uuid": "DEVICE-641b963e-aa61-46f8-a036-12023768427a",
    "Device group": "LTO",
    "World wide name": "ABCDEF1234567890",
    "Sym alias": "satGroup"
  },
  {
    "Description": "salesDivisionDrive",
    "Serial Number": "FAA49403AQJ1",
    "Device uuid": "DEVICE-4c3d0117-79a7-411e-a568-83481adc8332",
    "Device group": "LTO",
    "World wide name": "ABCDEF1234567891",
    "Sym alias": "satGroup"
  },
  {
    "Description": "salesDivisionDrive",
    "Serial Number": "FAA49403AQJ2",
    "Device uuid": "DEVICE-1fe2e310-6c2e-471f-96fd-838f03ac0d5d",
    "Device group": "LTO",
    "World wide name": "ABCDEF1234567892",
    "Sym alias": "satGroup"
  },
  {
    "Description": "salesDivisionDrive",
    "Serial Number": "FAA49403AQJ3",
    "Device uuid": "DEVICE-7890b61a-398c-4e61-afc0-8007a7274bc9",
    "Device group": "LTO",
        "World wide name": "ABCDEF1234567893",
        "Sym alias": "satGroup"
      },
    ]
```

Error response

```
Status Code : 400 Bad Request
Content-Language: en
{"code" "CTGKM6002E"
 , "message": "CTGKM6002E Bad Request: Invalid user authentication ID or
invalid request format."
 }
```

Device types

You must specify the appropriate device type in **`Device List Type REST Service`** to obtain a list of the device groups.

Step Returns a list of device groups for the **Guided key and device creation** list in the IBM Security Key Lifecycle Manager welcome page to select a device group.

Note: The following device families are excluded:

- `INTERNAL`
- `GENERIC`

Admin

Returns a list of device groups for the **Manage keys and devices** list in the IBM Security Key Lifecycle Manager welcome page to manage keys or certificates.

Note: The following device families are excluded:

- `INTERNAL`
- `GENERIC`

Rollover

Returns a list of device groups for the **Manage default rollover** list in the IBM Security Key Lifecycle Manager welcome page to specify keys or certificates to use on a future date.

Note: The following device families are excluded:

- DS8000
- DS5000
- INTERNAL
- GENERIC

IBM Security Key Lifecycle Manager includes the following device families:

- INTERNAL
- TS3592
- LTO
- DS5000
- GENERIC

Note: Device list contains all the base device groups such as DS5000, DS8000, LTO, and 3592 and also the user-defined device groups with the required user permissions.

Device List Type REST Service

Use **Device List Type REST Service** to obtain a list of device groups for the device type you specified.

Defined in

"Device types" on page 100

Operation

`Get`

URL

To retrieve a list for all device types:

`https://<host>:<port>/SKLM/rest/v1/deviceTypes`

To retrieve a list for a single device type:

`https://<host>:<port>/SKLM/rest/v1/deviceTypes<deviceListType>`

To retrieve a list for multiple device types:

`https://<host>:<port>/SKLM/rest/v1/deviceTypes?deviceListType=<value>`

Request

Request parameters

Parameter	Description
host	Specify the IP address or host name of the IBM Security Key Lifecycle Manager server.
port	Specify the Port number on which the IBM Security Key Lifecycle Manager server listens for requests.
deviceListType	Optional. Specify the device list type for which the list of device groups are to be returned. Device lists for all the types are returned if you do not specify this parameter. You can specify any of the following values: • admin • rollover • step You can specify multiple comma-separated values.

Request Headers

Header name	Value
Content-Type	application/json
Accept	application/json
Authorization	SKLMAuth userAuthId=<authIdValue>
Accept-Language	Any valid locale that is supported by IBM Security Key Lifecycle Manager. For example: en or de

Response

Response Headers

Header name	Value and description
Status Code	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.
Content-Type	application/json
Content-Language	Locale for the response message.

Success response body

JSON Object with the following specification:

JSON property name	Description
deviceListType (admin, step, rollover)	Returns the JSON object of device list type, such as admin, step, or rollover. The JSON object contains a list of device groups with JSON device object as specified in the following JSON device object table.

JSON device object

JSON property name	Description
name	Contains the device group name.
label	Contains the device group label name.

Error Response Body

JSON object with the following specification:

JSON property name	Description
code	Returns the application error code.
message	Returns a message that describes the error.

Examples

Service request to get device list of all types

```
GET https://<host>:<post>/SKLM/rest/v1/deviceTypes
Content-Type: application/json
Accept : application/json
Authorization : SKLMAuth userAuthId=37ea1939-1374-4db7-84cd-14e399be2d20
Accept-Language : en
```

Success response

```
Status Code : 200 OK
Content-Language: en
{"admin":
{"items":
    [{ ''name'':''LTO001'',
    ''label'':''1374-4db7-84cd-14e399be2d20''
    },
    { ''name'':"DISK002'',
    ''label'':''1374-4db7-84cd-14e399be2d21''
    }],"identifier":"name"},
"rollover":
{"items":
    [{''name'':''LTO001'',
    ''label'':''1374-4db7-84cd-14e399be2d20''
    },
    { ''name'':"DISK002'',
    ''label'':''1374-4db7-84cd-14e399be2d21''
    }],"identifier":"name"},
"step":
{"items":
    [{''name'':''LTO001'',
    ''label'':''1374-4db7-84cd-14e399be2d20''
    },
    { ''name'':"DISK002'',
    ''label'':''1374-4db7-84cd-14e399be2d21''
    }],"identifier":"name"}
}
```

Service request to get a list of single device type

```
GET https://<host>:<post>/SKLM/rest/v1/deviceTypes?deviceListType=admin
OR
GET https://<host>:<post>/SKLM/rest/v1/deviceTypes/<deviceListType>
Content-Type: application/json
Accept : application/json
Authorization : SKLMAuth userAuthId=37ea1939-1374-4db7-84cd-14e399be2d20
Accept-Language : en
```

Success response

```
Status Code : 200 OK
Content-Language: en
{"admin":
{"items":
    [{ ''name'':''LTO001'',
    ''label'':''1374-4db7-84cd-14e399be2d20''
    },
    { ''name'':"DISK002'',
    ''label'':''1374-4db7-84cd-14e399be2d21''
    }],"identifier":"name"}
}
```

Service request to get list of multiple device types

```
GET https://<host>:<post>/SKLM/rest/v1/deviceTypes?deviceListType=admin,step
Content-Type: application/json
Accept : application/json
Authorization : SKLMAuth userAuthId=37ea1939-1374-4db7-84cd-14e399be2d20
Accept-Language : en
```

Success response

```
Status Code : 200 OK
Content-Language: en
{"admin":
{"items":
    [{ ''name'':''LTO001'',
    ''label'':''1374-4db7-84cd-14e399be2d20''
    },
    { ''name'':"DISK002'',
    ''label'':''1374-4db7-84cd-14e399be2d21''
    }],"identifier":"name"},
"step":
{"items":
    [{''name'':''LTO001'',
    ''label'':''1374-4db7-84cd-14e399be2d20''
    },
    { ''name'':"DISK002'',
    ''label'':''1374-4db7-84cd-14e399be2d21''
    }],"identifier":"name"}
```

Error response

```
Status Code : 400 Bad Request
Content-Language: en
{"code" : "CTGKM6002E", "message" : "CTGKM6002E Bad Request: Invalid
user authentication ID or invalid request format."}
```

Device Update REST Service

Use **`Device Update REST Service`** to update the attributes of a device in the IBM Security Key Lifecycle Manager database.

Operation

`PUT`

URL `https://<host>:<port>/SKLM/rest/v1/devices`

Request

Request Parameters

Parameter	Description
host	Specify the IP address or host name of the IBM Security Key Lifecycle Manager server.
port	Specify the port number on which the IBM Security Key Lifecycle Manager server listens for requests.

Request Headers

Header name	Value
Content-Type	`application/json`
Accept	`application/json`
Authorization	`SKLMAuth userAuthId=<authIdValue>`
Accept-Language	Any valid locale that is supported by IBM Security Key Lifecycle Manager. For example: `en` or `de`

Request body

JSON object with the following specification:

Property name	Description
uuid	Required. Specifies the universal unique identifier of the device. For example, `DEVICE-74386920-148c-47b2-a1e2-d19194b315cf`.

Request body

JSON object with the following specification:

Property name	Description
type	Specify a unique device group, such as myNewLTO. You can specify any of the following values: **LTO** Specifies the LTO device group. **3592** Specifies the 3592 device group. **DS5000** Specifies the DS5000 device group. **DS8000** Specifies the DS8000 device group. **BRCD_ENCRYPTOR** Specifies the BRCD_ENCRYPTOR device group that is in the LTO device family. **ONESECURE** Specifies the ONESECURE device group that is in the DS5000 device family. **GENERIC** Specifies a device family that uses the Key Management Interoperability Protocol to interact with IBM Security Key Lifecycle Manager. The GENERIC device group enables management of KMIP objects. Do not use the REST interface to add a device to the GENERIC device group or to change a GENERIC device group attribute. **ETERNUS_DX** Specifies the ETERNUS_DX device group that is in the DS5000 device family. **XIV** Specifies the XIV device group that is in the DS5000 device family. **userdevicegroup** Specifies a user-defined group that is based on a supported device family.
attributes	Specify one or more attribute-value pairs. The values are stored in the IBM Security Key Lifecycle Manager database. You can specify any of the following values:

Request body

JSON object with the following specification:

Property name	Description
	aliasOne Specifies a default alias for a certificate that is used by a 3592 tape drive or a DS5000 turbo drive. Not used for an LTO tape drive or DS5000 storage server. - 3592 tape drive The value is optional for a 3592 tape drive. Specifies the primary certificate that the device in the 3592 device family uses if the primary certificate is not available. If this attribute is not specified, the partner default certificate is used. The certificate is specified as a table entry for the device group in the IBM Security Key Lifecycle Manager database. - DS8000 turbo drive The value is optional for a DS8000 turbo drive. Matches the label "Primary certificate for image" in the graphical user interface panels for a DS8000 Turbo drive. Use **Device Group Attribute List REST Service** and **Device Group Attribute Update REST Service** to view and change the value. This value was previously stored in the **drive.default.alias1** configuration parameter. **aliasTwo** Used for a 3592 tape drive or a DS8000 turbo drive. Not used for an LTO tape drive or DS5000 storage server. - 3592 tape drive This attribute specifies a default alternative alias for a 3592 tape drive. This value can be the same, or different from the value that is specified for the primary certificate. The value specifies the secondary certificate that the device in the 3592 device family uses if the primary certificate is not available. If this attribute is not specified, the partner default certificate is used. The certificate is specified as a table entry for the device group in the IBM Security Key Lifecycle Manager database. - DS8000 turbo drive For a device in the DS8000 device family, the value specifies a secondary certificate that is available for use. For example, you might use this certificate to unlock a DS8000 turbo drive in the case of a deadlock condition. Use **Device Group Attribute List REST Service** and **Device Group Attribute Update REST Service** to view and change the value. This value was previously stored in the **drive.default.alias2** configuration parameter.

Request body

JSON object with the following specification:

Property name	Description
	description Specifies more information that describes the type of device or its purpose. **deviceText** Optional. Specifies the unique text with a minimum length greater than zero bytes and a maximum length of 96 bytes that describes a DS5000 storage server. **serialNumber** For a DS5000 storage server, specifies the serial number of drive. You can change the serial number of a DS5000 storage server to another serial number that is not currently in use. **symAlias** Specifies an alias that is used to identify an existing key group for an LTO tape drive. The attribute is also used for DS5000 storage server to change or associate a new device key container. This value is stored in the IBM Security Key Lifecycle Manager database. **worldwideName** Specifies the name of a device. This name is a nonsecure address that is used in combination with other device information, such as a serial number, to define devices and device paths. Specify a 16-character hexadecimal value that contains only the characters `ABCDEFabcdef1234567890`.

Response

Response Headers

Header name	Value and description
`Status Code`	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.
`Content-Type`	`application/json`
`Content-Language`	Locale for the response message.

Success response body

JSON object with the following specification:

JSON property name	Description
code	Returns an integer value such as 0 or 1 to indicate the device update status.
status	Returns the status message to indicate whether the device update task was successful. **0** The device update task succeeded. **1** The device is not updated.

Error Response Body

JSON object with the following specification:

JSON property name	Description
code	Returns the application error code.
message	Returns a message that describes the error.

Examples

Service request to update a device

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

Success response

```
Status Code: 200 OK
{"code": "0","status": "Succeeded"}
```

Service request to update a device when attribute value is not specified

```
PUT https://localhost:9080/SKLM/rest/v1/devices
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
{"uuid":"DEVICE-80896a45-8f9e-428d-a3ac-ac9ffcbbfe8b","attributes":
"description "}
```

Success response

```
Status Code: 200 OK
{"code":"1","status":"CTGKM0509I No update"}
```

Error response when `uuid` is not specified

```
PUT https://localhost:9080/SKLM/rest/v1/devices
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
{"uuid":""}
```

Error response

```
Status Code: 400 Bad Request
{"code":"CTGKM0550E","message":"CTGKM0550E Input value cannot be an
empty string for parameter uuid "}}
```

Get Config Properties List REST Service

Use **Get Config Properties List REST Service** to retrieve a list of all properties from the IBM Security Key Lifecycle Manager configuration properties file. You can view and change the configuration properties.

Operation
GET

URL `https://<host>:<port>/SKLM/rest/v1/configProperties`

Request

Request Parameters

Parameter	Description
host	Specify the IP address or host name of the IBM Security Key Lifecycle Manager server.
port	Specify the port number on which the IBM Security Key Lifecycle Manager server listens for requests.

Request Headers

Header name	Value
Content-Type	application/json
Accept	application/json
Authorization	SKLMAuth userAuthId=<authIdValue>
Accept-Language	Any valid locale that is supported by IBM Security Key Lifecycle Manager. For example: en or de

Response

Response Headers

Header name	Value and description
Status Code	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.
Content-Type	application/json
Content-Language	Locale for the response message.

Success response body

JSON Object with the following specification:

JSON property name	Description
configProperty	Returns the JSON object that contains the properties. Each property includes the property name and its value.

Error Response Body

JSON object with the following specification:

JSON property name	Description
code	Returns the application error code.
message	Returns a message that describes the error.

Examples

Service request to list all the configuration properties

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

Success response

```
Status Code : 200 OK
Content-Language: en
 {
       "KMIPListener.ssl.port": "5696",
       "TransportListener.ssl.timeout": "10",
       "Audit.handler.file.size": "10000",
       "config.keystore.name": "defaultKeyStore",
       "tklm.encryption.password": "******",
       "debug.output.file": "logs/debug/sklm_debug.log",
       "rest.user.inactive_time": "15",
       "debug.output": "simple_file",
       "Audit.event.types": "runtime,authorization,authorization_terminate,
    resource_management,key_management",
       "enableKeyRelease": "false",
       "TransportListener.tcp.port": "3801",
       "Audit.eventQueue.max": "0",
       "config.keystore.batchUpdateTimer": "60000",
       "Audit.handler.file.name": "logs/audit/sklm_audit.log",
       "enableClientCertPush": "false",
       "debug": "all",
       "TransportListener.tcp.timeout": "10",
       "backup.keycert.before.serving": "false",
       "TransportListener.ssl.protocols": "SSL_TLS",
       "cert.valiDATE": "false",
       "config.keystore.batchUpdateSize": "10000",
       "useSKIDefaultLabels": "false",
       "config.keystore.ssl.certalias": "test-key",
       "TransportListener.ssl.port": "441",
       "fips": "off",
       "zOSCompatibility": "false",
       "Audit.event.outcome": "success,failure"
    }
```

Error response

```
Status Code : 400 Bad Request
Content-Language: en
{"code" : "CTGKM6002E", "message" : "CTGKM6002E Bad Request: Invalid user
authentication ID or invalid request format."}
```

Get Multiple Config Properties REST Service

Use **Get Multiple Config Properties REST Service** to retrieve one or more properties from the SKLMConfig.properties configuration file. You can view and change the configuration properties.

Operation
GET

URL https://<*host*>:<*port*>/SKLM/rest/v1/configProperties?properties=<propertyName1>,<propertyName2>,<...>

Request

Request Parameters

Parameter	Description
host	Specify the IP address or host name of the IBM Security Key Lifecycle Manager server.
port	Specify the port number on which the IBM Security Key Lifecycle Manager server listens for requests.

Request Headers

Header name	Value
Content-Type	application/json
Accept	application/json
Authorization	SKLMAuth userAuthId=<authIdValue>
Accept-Language	Any valid locale that is supported by IBM Security Key Lifecycle Manager. For example: en or de

Query parameters

JSON property name	Description
properties	Specify the configuration property names that you want to retrieve. You can specify multiple comma-separated properties.

Response

Response Headers

Header name	Value and description
Status Code	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.
Content-Type	application/json
Content-Language	Locale for the response message.

Success response body

JSON array that contains JSON objects with the following specification:

JSON property name	Description
property	Returns the name of the requested configuration property.
value	Returns the value of the configuration property. This JSON property is present only when the property is found in the configuration file.
status	Returns the status to indicate that the requested property does not exist. This JSON property is present only when the property is not found in the configuration file.

Note: The success response code 200 OK is returned even if the property you requested is not found. An appropriate message is returned in the response body.

Error Response Body

JSON object with the following specification:

JSON property name	Description
code	Returns the application error code.
message	Returns a message that describes the error.

Examples

Service request to get a single configuration property

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

Success response

```
Status Code : 200 OK
Content-Language: en
[{"property":"KMIPListener.ssl.port","value":"5696"}]
```

Error response

```
Status Code : 400 Bad Request
Content-Language: en
{"code" : "CTGKM6002E", "message" : "CTGKM6002E Bad Request: Invalid
user authentication ID or invalid request format."}
```

Service request to get multiple configuration properties

```
GET https://localhost:9080/SKLM/rest/v1/configProperties?properties=
KMIPListener.ssl.port,ghtj,TransportListener.ssl.timeout
Content-Type: application/json
Accept : application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
Accept-Language : en
```

Success response

```
Status Code : 200 OK
Content-Language: en
[
      {
          "property": "KMIPListener.ssl.port",
          "value": "5696"
      },
      {
          "property": "ghtj",
          "status": "CTGKM0556E Cannot find the property in
configuration file: ghtj "
      },
      {
          "property": "TransportListener.ssl.timeout",
          "value": "10"
      }
 ]
```

Error response

```
Status Code : 400 Bad Request
Content-Language: en
{"code" : "CTGKM6002E", "message" : "CTGKM6002E Bad Request: Invalid
user authentication ID or invalid request format."}
```

Get Multiple Replication Config Properties REST Service

Use **Get Multiple Replication Config Properties REST Service** to retrieve one or more properties from the ReplicationSKLMConfig.properties configuration file.

Operation
GET

URL https://<*host*>:<*port*>/SKLM/rest/v1/replicationConfigProperties?properties=<propertyName1>,<propertyName2>,<....>

Request

Request Parameters

Parameter	Description
host	Specify the IP address or host name of the IBM Security Key Lifecycle Manager server.
port	Specify the port number on which the IBM Security Key Lifecycle Manager server listens for requests.

Request Headers

Header name	Value
Content-Type	application/json
Accept	application/json
Authorization	SKLMAuth userAuthId=<authIdValue>
Accept-Language	Any valid locale that is supported by IBM Security Key Lifecycle Manager. For example: en or de

Query parameters

JSON property name	Description
properties	Specify the replication configuration property names that you want to retrieve. You can specify multiple comma-separated property names.

Response

Response Headers

Header name	Value and description
Status Code	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.
Content-Type	application/json
Content-Language	Locale for the response message.

Success response body

JSON object with the following specification:

JSON property name	Description
property	Returns the replication configuration property name that you requested.
value	Returns the value of the replication configuration property. This JSON property is present only when the configuration property is found in the replication configuration file.
status	Returns the status to indicate that the requested property does not exist. This JSON property is present only when the configuration property is found in the replication configuration file.

Note: The success response code `200 OK` is returned even if the property you requested is not found. An appropriate message is returned in the response body.

Error Response Body

JSON object with the following specification:

JSON property name	Description
code	Returns the application error code.
message	Returns a message that describes the error.

Examples

Service request to return a single configuration property

```
GET https://localhost:9080/SKLM/rest/v1/replicationConfigProperties?
properties=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
[{"property":"replication.role","value":"master"}]
```

Error response

```
Status Code : 400 Bad Request
Content-Language: en
{"code" : "CTGKM6002E", "message" : "CTGKM6002E Bad Request: Invalid
user authentication ID or invalid request format."}
```

Service request to return multiple configuration properties

```
GET https://localhost:9080/SKLM/rest/v1/replicationConfigProperties?
propertyName=replication.role,ghtj,backup.ClientPort1
Content-Type: application/json
Accept : application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
Accept-Language : en
```

Success response

```
Status Code : 200 OK
Content-Language: en
[
       {
          "property":"replication.role",
          "value":"master"
       },
       {
          "property": "ghtj",
          "status": "CTGKM0556E Cannot find the property in
           configuration file: ghtj "
       },
       {
          "property": "backup.ClientPort1",
          "value": "1024"
       }
    ]
```

Error response

```
Status Code : 400 Bad Request
Content-Language: en
{"code" : "CTGKM6002E", "message" : "CTGKM6002E Bad Request: Invalid
user authentication ID or invalid request format."}
```

Get Single Config Property REST Service

Use **Get Single Config Property REST Service** to retrieve a specific property from the SKLMConfig.properties configuration file.

Operation
GET

URL https://*<host>*:*<port>*/SKLM/rest/v1/configProperties/{propertyName}

Request

Request Parameters

Parameter	Description
host	Specify the IP address or host name of the IBM Security Key Lifecycle Manager server.
port	Specify the port number on which the IBM Security Key Lifecycle Manager server listens for requests.

Request Headers

Header name	Value
Content-Type	application/json
Accept	application/json
Authorization	SKLMAuth userAuthId=<authIdValue>
Accept-Language	Any valid locale that is supported by IBM Security Key Lifecycle Manager. For example: en or de

Path parameters

Property name	Description
propertyName	Specifies the name of the configuration property for which you want to retrieve the value.

Response

Response Headers

Header name	Value and description
Status Code	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.
Content-Type	application/json
Content-Language	Locale for the response message.

Success response body

JSON object with the following specification:

JSON property name	Description
configProperty	JSON object that contains the properties. Each property represents the property name and its value.

Error Response Body

JSON object with the following specification:

JSON property name	Description
code	Returns the application error code.
message	Returns a message that describes the error.

Examples

Service request to get a specific configuration property

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

Success response

```
Status Code : 200 OK
Content-Language: en
{" KMIPListener.ssl.port " : "5696"}
```

Error response

```
Status Code : 400 Bad Request
Content-Language: en
{"code" : "CTGKM6002E", "message" : "CTGKM6002E Bad Request: Invalid user
authentication ID or invalid request format."}
```

Get Single Replication Config Properties REST Service

Use **Get Single Replication Config Properties REST Service** to retrieve a specific property from the ReplicationSKLMConfig.properties configuration file.

Operation
GET

URL https://*<host>*:*<port>*/SKLM/rest/v1/replicationConfigProperties/{propertyName}

Request

Request Parameters

Parameter	Description
host	Specify the IP address or host name of the IBM Security Key Lifecycle Manager server.
port	Specify the port number on which the IBM Security Key Lifecycle Manager server listens for requests.

Request Headers

Header name	Value
Content-Type	application/json
Accept	application/json
Authorization	SKLMAuth userAuthId=<authIdValue>
Accept-Language	Any valid locale that is supported by IBM Security Key Lifecycle Manager. For example: en or de

Path parameters

Property name	Description
<propertyName>	Specify the name of the replication configuration property for which you want to retrieve the value.

Response

Response Headers

Header name	Value and description
Status Code	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.
Content-Type	application/json
Content-Language	Locale for the response message.

Success response body

JSON object with the following specification:

JSON property name	Description
<configProperty>	Returns the JSON object that contains the property. It represents configuration property name and its value.

Error Response Body

JSON object with the following specification:

JSON property name	Description
code	Returns the application error code.
message	Returns a message that describes the error.

Examples

Service request to return a single configuration property

```
GET https://localhost:9080/SKLM/rest/v1/replicationConfigProperties/
backup.ClientPort1
Content-Type: application/json
Accept : application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
Accept-Language : en
```

Success response

```
Status Code : 200 OK
Content-Language: en
{"property" : "backup.ClientPort1","value" : "1024"}
```

Error response

```
Status Code : 400 Bad Request
Content-Language: en
{"code" : "CTGKM6002E", "message" : "CTGKM6002E Bad Request: Invalid user
authentication ID or invalid request format."}
```

Get Replication Config Properties List REST Service

Use **Get Replication Config Properties List REST Service** to retrieve a list of all configuration properties from the ReplicationSKLMConfig.properties configuration file.

Operation
GET

URL https://*<host>*:*<port>*/SKLM/rest/v1/replicationConfigProperties

Request

Request Parameters

Parameter	Description
host	Specify the IP address or host name of the IBM Security Key Lifecycle Manager server.
port	Specify the port number on which the IBM Security Key Lifecycle Manager server listens for requests.

Request Headers

Header name	Value
Content-Type	application/json
Accept	application/json
Authorization	SKLMAuth userAuthId=<authIdValue>
Accept-Language	Any valid locale that is supported by IBM Security Key Lifecycle Manager. For example: en or de

Response

Response Headers

Header name	Value and description
Status Code	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.
Content-Type	`application/json`
Content-Language	Locale for the response message.

Success response body

JSON object with the following specification:

JSON property name	Description
<configProperty>	Returns the JSON object that contains a list of comma-separated replication properties. Each property represents the property name and its value.

Error Response Body

JSON object with the following specification:

JSON property name	Description
code	Returns the application error code.
message	Returns a message that describes the error.

Examples

Service request to return a list of all properties

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

Success response

```
Status Code : 200 OK
Content-Language: en
{ "backup.ClientPort1:1024, replication.role:master,replication.
MaxLogFileSize:2000,replication.auditLogName:test.log"}
```

Error response

```
Status Code : 400 Bad Request
Content-Language: en
{"code" : "CTGKM6002E", "message" : "CTGKM6002E Bad Request: Invalid user
authentication ID or invalid request format."}
```

Get System Details REST Service

Use **Get System Details REST Service** to request IBM Security Key Lifecycle Manager server configuration details.

You can use **Get System Details REST Service** to get the following details:

- IP address of the server
- Keystore initialization status
- TCP transports initialization status
- TCP port number
- SSL configuration status
- SSL port number
- KMIP implementation status
- KMIP SSL port number of the server
- KMIP initialization status

Operation

`GET`

URL

To retrieve all system details:

```
https://<host>:<port>/SKLM/rest/v1/systemDetails
```

To retrieve a single system detail:

```
https://<host>:<port>/SKLM/rest/v1/systemDetails/
<systemDetailType>
```

To retrieve multiple system details:

```
https://<host>:<port>/SKLM/rest/v1/
systemDetails?systemDetailType=<value>
```

Request

Request parameters

Parameter	Description
host	Specify the IP address or host name of the IBM Security Key Lifecycle Manager server.
port	Specify the port number on which the IBM Security Key Lifecycle Manager server listens for requests.

Request parameters

Parameter	Description
systemDetailType	Optional. Specify the configuration detail of the IBM Security Key Lifecycle Manager server. All the configuration details of the server are returned if this parameter is not specified. You can specify any of the following values: • IPAddr • keyStoreInitialized • TCPTransportsInitialized • SSLTransportsInitialized • TCPport • SSLConfigured • SSLport • KMIPConfigured • KMIPSSLport • KMIPStatus You can specify multiple comma-separated values.

Request Headers

Header name	Value
Content-Type	application/json
Accept	application/json
Authorization	SKLMAuth userAuthId=<authIdValue>
Accept-Language	Any valid locale that is supported by IBM Security Key Lifecycle Manager. For example: en or de

Response

Response Headers

Header name	Value and description
Status Code	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.
Content-Type	application/json
Content-Language	Locale for the response message.

Success response body

JSON object with the following specification:

JSON property name	Description
IPAddr	Returns the IP address or host name of the IBM Security Key Lifecycle Manager server.
keyStoreInitialized	Indicates whether the keystore is initialized in the server.
TCPTransportsInitialized	Indicates whether the TCP transport is initialized for message communications.
SSLTransportsInitialized	Indicates whether the SSL transport is initialized in the server for secure communications.
TCPport	Returns the TCP port number of the IBM Security Key Lifecycle Manager server.
SSLConfigured	Indicates whether SSL is configured for the IBM Security Key Lifecycle Manager server to use an SSL connection
SSLport	Returns the SSL port number on which the server listens for SSL requests.
KMIPConfigured	Indicates whether the KMIP settings are configured in the IBM Security Key Lifecycle Manager server.
KMIPSSLport	Returns the port number on which theIBM Security Key Lifecycle Manager server listens for requests to communicate over the SSL socket by using KMIP.
KMIPStatus	Indicates whether KMIP is used for key management operations by the IBM Security Key Lifecycle Manager server.

Error Response Body

JSON object with the following specification:

JSON property name	Description
code	Returns the application error code.
message	Returns a message that describes the error.

Examples

Service request to get all system details

```
GET https://<host>:<port>/SKLM/rest/v1/systemDetails
Content-Type: application/json
Accept : application/json
Authorization : SKLMAuth userAuthId=37ea1939-1374-4db7-84cd-14e399be2d20
Accept-Language : en
```

Success response

```
Status Code : 200 OK
Content-Language: en
{
"TCPPort": "3801",
"keyStoreInitialized": false,
"KMIPSSLport": "5696",
"SPIKEConfigured": false,
"KMIPConfigured": true,
"IPAddr": "9.118.41.254",
"TCPTransportsInitialized": false,
"SPIKECert": "",
"SSLConfigured": true,
"KMIPStatus": false,
"SSLPort": "441",
"SSLTransportsInitialized": false
}
```

Service request to get a single system detail

```
GET https://<host>:<port>/SKLM/rest/v1/systemDetails?systemDetailType=IPAddr
OR
GET https://<host>:<port>/SKLM/rest/v1/systemDetails/IPAddr
Content-Type: application/json
Accept : application/json
Authorization: SKLMAuth userAuthId=37ea1939-1374-4db7-84cd-14e399be2d20
Accept-Language: en
```

Success response

```
Status Code : 200 OK
Content-Language: en
{"IPAddr" : "9.118.41.254"}
```

Service request to get multiple system details

```
GET https://<host>:<port>/SKLM/rest/v1/systemDetails?systemDetailType=
TCPTransportsInitialized,TCPport
Content-Type: application/json
Accept : application/json
Authorization : SKLMAuth userAuthId=37ea1939-1374-4db7-84cd-14e399be2d20
Accept-Language : en
```

Success response

```
Status Code : 200 OK
Content-Language: en
{"TCPTransportsInitialized" : true,"TCPport" : "8080"}
```

Error Responses

```
Status Code : 400 Bad Request
Content-Language: en
{"code" : "CTGKM6002E", "message" : "CTGKM6002E Bad Request: Invalid
user authentication ID or invalid request format."}
```

Group Create REST Service

Use **Group Create REST Service** to create a key group to which you can add keys.

Operation
POST

URL https://*<host>*:*<port>*/SKLM/rest/v1/keygroups/{groupName}

Request

Request Parameters

Parameter	Description
host	Specify the IP address or host name of the IBM Security Key Lifecycle Manager server.
port	Specify the port number on which the IBM Security Key Lifecycle Manager server listens for requests.

Request Headers

Header name	Value
Content-Type	application/json
Accept	application/json
Authorization	SKLMAuth userAuthId=<authIdValue>

Request Headers

Header name	Value
Accept-Language	Any valid locale that is supported by IBM Security Key Lifecycle Manager. For example: en or de

Path parameter

Parameter	Description
groupName	Specify a unique name of the group to be created.

Request body

JSON object with the following specification:

JSON property name	Description
usage	Specify the device to which this key group is associated, such as LTO.

Response

Response Headers

Header name	Value and description
Status Code	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.
Content-Type	application/json
Content-Language	Locale for the response message.

Success response body

JSON object with the following specification:

JSON property name	Description
GroupUuid	Returns the unique ID of the newly created key group.

Error Response Body

JSON object with the following specification:

JSON property name	Description
code	Returns the application error code.
message	Returns a message that describes the error.

Examples

Service request to create a group

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

Success response

```
Status Code: 200 OK
{"GroupUuid":"KEYGROUP-36092084-4fe9-4a01-b766-906bb293317a"}
```

Error response

```
Status Code: 400 Bad Request
{"code":"CTGKM6002E","message":"CTGKM6002E Bad Request: Invalid user
authentication ID or invalid request format."}
```

```
Status Code: 500 Internal Server Error
{"code":"CTGKM0583E","message":"CTGKM0583E Group already exists:
newGroup "}
```

Group Delete REST Service

Use **Group Delete REST Service** to delete a key group. Deleting a populated key group also deletes all the keys in the key group.

Operation
`DELETE`

URL `https://<host>:<port>/SKLM/rest/v1/keygroups/<uuid>`

Request

Request Parameters

Parameter	Description
host	Specify the IP address or host name of the IBM Security Key Lifecycle Manager server.
port	Specify the port number on which the IBM Security Key Lifecycle Manager server listens for requests.

Request Headers

Header name	Value
Content-Type	application/json
Accept	application/json
Authorization	SKLMAuth userAuthId=<authIdValue>
Accept-Language	Any valid locale that is supported by IBM Security Key Lifecycle Manager. For example: en or de

Path parameters

Parameter name	Description
uuid	Specify a unique identifier for the group. An example of a key group UUID is GROUP-7d588437-e725-48bf-a836-00a47df64e78.

Response

Response Headers

Header name	Value and description
Status Code	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.
Content-Type	application/json
Content-Language	Locale for the response message.

Success response body

JSON object with the following specification:

JSON property name	Description
status	Returns the status to indicate the deletion of specified key group.

Error Response Body

JSON object with the following specification:

JSON property name	Description
code	Returns the application error code.
message	Returns a message that describes the error.

Examples

Service request to delete a key group

```
https://localhost:9080/SKLM/rest/v1/keygroups/KEYGROUP-720aac61-9516-4bb9-
9fa3-88f5db2823db
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
```

Success response

```
Status Code: 200 OK
{"code":"0","status":"Succeeded"}
```

Error response

```
Status Code: 400 Bad Request
{"code":"CTGKM6002E","message":"CTGKM6002E Bad Request: Invalid user
authentication ID or invalid request format."}
```

Group Entry Add REST Service

Use **Group Entry Add REST Service** to add keys to an existing key group. Your role must have a permission to the modify action and a permission to the appropriate device group.

Operation
`POST`

URL `https://<host>:<port>/SKLM/rest/v1/keygroupentry`

Request

Request Parameters

Parameter	Description
host	Specify the IP address or host name of the IBM Security Key Lifecycle Manager server.
port	Specify the port number on which the IBM Security Key Lifecycle Manager server listens for requests.

Request Headers

Header name	Value
Content-Type	application/json
Accept	application/json
Authorization	SKLMAuth userAuthId=<authIdValue>
Accept-Language	Any valid locale that is supported by IBM Security Key Lifecycle Manager. For example: en or de

Request body

JSON object with the following specification:

Parameter name	Description
entry	Required. Specify the entry to add to an existing key group. You can specify a comma-separated key-value pair. Any one of the following attributes is required: **alias** Unique name of the existing key. **uuid** Unique universal identifier of an entry. For example, KEY-a3ce9230-bef9-42bd-86b7-6d208ec119cf. If you specify value for both the attributes, value of **uuid** is considered.
name	Required. Specify a unique name of the existing group.

Response

Response Headers

Header name	Value and description
Status Code	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.
Content-Type	application/json
Content-Language	Locale for the response message.

Success response body

JSON object with the following specification:

JSON property name	Description
code	Returns the code that is specified by the **status** property.
status	Returns the status to indicate whether addition of key to the group was successful.

Error Response Body

JSON object with the following specification:

JSON property name	Description
code	Returns the application error code.
message	Returns a message that describes the error.

Examples

Service request for adding a key to an existing group

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

Success response

```
Status Code : 200 OK
{"code":"0","status":"Succeeded"}
```

Service request to add a key when an incorrect entry is specified

```
POST https://localhost:9080/SKLM/rest/v1/keygroupentry
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
{"name":"myGroup", "entry": "alias aaa000000000000000002"}
```

Error response

```
Status Code : 500 Internal Server Error
{"code":"CTGKM0565E","message":"CTGKM0565E Cannot find the key:
aaa000000000000000002"}
```

Group List REST Service

Use **Group List REST Service** to list the objects in a group of keys, or the groups of a specific type.

Group List REST Service supports pagination. The request parameters, such as offset and count, are used for pagination. For example, to retrieve the first 10 records for the list, set **offset = 1** and **count = 10**. To retrieve the next 10 records, set **offset = 2** and **count = 10**. If you do not specify values for pagination parameters, the first 2000 records are returned.

Operation

GET

URL

To retrieve all keys:

https://*<host>*:*<port>*/SKLM/rest/v1/keygroups

Note: Returns the first 2000 records.

To retrieve a specific list with pagination:

https://*<host>*:*<port>*/SKLM/rest/v1/keygroups?offset=<offset>&count=<count>

Request

Request Parameters

Parameter	Description
host	Specify the IP address or host name of the IBM Security Key Lifecycle Manager server.
port	Specify the port number on which the IBM Security Key Lifecycle Manager server listens for requests.

Request Headers

Header name	Value
Content-Type	application/json
Accept	application/json
Authorization	SKLMAuth userAuthId=<authIdValue>
Accept-Language	Any valid locale that is supported by IBM Security Key Lifecycle Manager. For example: en or de

Query parameters

Parameter name	Description
name	Specify the group name.

Query parameters

Parameter name	Description
usage	Specify the unique device group, such as LTO. You can specify any of the following values: **LTO** Specifies the LTO device group. **DS5000** Specifies the DS5000 device group. **BRCD_ENCRYPTOR** Specifies the BRCD_ENCRYPTOR device group that is in the LTO device family. **ONESECURE** Specifies the ONESECURE device group that is in the DS5000 device family. **GENERIC** Specifies a device family that uses the Key Management Interoperability Protocol to interact with Security Key Lifecycle Manager. The GENERIC device group enables management of KMIP objects. **ETERNUS_DX** Specifies the ETERNUS_DX device group that is in the DS5000 device family. **XIV** Specifies the XIV device group that is in the DS5000 device family. **userdevicegroup** Specifies a user-defined group that is based on a supported device family.
offset	Specify the page number from which the records are displayed based on the value that you specify for **count**.
count	Specify the number of records to display on the page that you specified with **offset**. The count must not exceed 2000 records.

Response

Response Headers

Header name	Value and description
Status Code	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.
Content-Type	application/json

Response Headers

Header name	Value and description
Content-Language	Locale for the response message.

Success response body

JSON object with the following specification:

JSON property name	Description
group name	Returns the name of the key group.
group type	Returns the type of the objects in the group.
group uuid	Returns the unique ID of the group.
initialization date	Returns the group creation date.
activation date	Returns the group activation date.
key	JSON array with JSON objects. Contains information about the keys, such as `uuid`, `aliases`, and, `key store name`.
certificate	JSON array with JSON objects. Contains information about the certificate, such as `uuid`, `aliases`, and, `key store name`.
device	JSON array with JSON objects. Contains information about the device, such as `uuid`, `serial number`, and `device group name`.

Error Response Body

JSON object with the following specification:

JSON property name	Description
code	Returns the application error code.
message	Returns a message that describes the error.

Examples

Service request to list the objects in a group of keys

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

Success response

```
Status Code : 200 OK
Content-Language: en
[
   {
       "group name": "keygroup1",
       "group type": "KEY",
       "group uuid": "KEYGROUP-b0992af0-ae98-4a89-9c3c-dcc30ad6a347",
       "initialization date": "12/20/12 1:05:45 PM Central Standard
        Time",
       "activation date": "12/20/12 1:05:45 PM Central Standard Time",
       "keys":
       [
         {
           "uuid": "KEY-d58c43dc-302a-42e0-8c7a-2543a7d19548",
           "alias(es)": "aaa000000000000000000 ",
           "key store name(s)": "defaultKeyStore "
         },
         {
           "uuid": "KEY-61bd4100-9880-450f-a1a5-7efe19a8d0f5",
           "alias(es)": "aaa000000000000000001 ",
           "key store name(s)": "defaultKeyStore "
        },
        {
           "uuid": "KEY-9469e4d6-b12b-48e1-a33c-a040c9d303f4",
           "alias(es)": "aaa000000000000000002 ",
           "key store name(s)": "defaultKeyStore "
        }
      ],
       "usage": "LTO"
   }
   {
       "group name": "newgrp1",
       "group type": "KEY",
       "group uuid": "KEYGROUP-1e54f750-0a81-43fa-8178-79588780c369",
       "initialization date": "3/7/13 7:40:03 AM Central Standard
        Time",
       "activation date": "3/7/13 7:40:03 AM Central Standard Time",
       "usage": "LTO"
   },
   {
       "group name": "Group1",
       "group type": "KEY",
       "group uuid": "KEYGROUP-4276f3f9-29cb-4af9-861f-e81afdfda44a",
       "initialization date": "3/7/13 11:59:16 AM Central Standard
        Time",
       "activation date": "3/7/13 11:59:16 AM Central Standard Time",
       "usage": "LTO"
   }
]
```

Group List with pagination

```
GET https://localhost:9080/SKLM/rest/v1/keygroups?offset=1&count=1&name=
newgrp
Content-Type: application/json
Accept : application/json Authorization: SKLMAuth userAuthId=139aeh34567m
Accept-Language : en
```

Success response

```
Status Code : 200 OK
Content-Language: en
    [
       {
           "group name": "newgrp",
           "group type": "KEY",
           "group uuid": "KEYGROUP-2238a83c-a4de-46d7-bc5a-6c24bf858bc5,
           "initialization date": "2/7/13 1:40:16 PM Central Standard
            Time",
           "activation date": "2/7/13 1:40:16 PM Central Standard Time",
           "usage": "LTO"
       }
    ]
```

Error response

```
Status Code : 400 Bad Request
Content-Language: en
{"code":"CTGKM6002E","message":"CTGKM6002E Bad Request: Invalid user
authentication ID or invalid request format."}
```

Group Update REST Service

Use **Group Update REST Service** to update group metadata in the database for moving all the keys in a key group from one device group to another, within the same device group family.

Operation
PUT

URL `https://<host>:<port>/SKLM/rest/v1/keygroups`

Request

Request Parameters

Parameter	Description
host	Specify the IP address or host name of the IBM Security Key Lifecycle Manager server.
port	Specify the port number on which the IBM Security Key Lifecycle Manager server listens for requests.

Request Headers

Header name	Value
Content-Type	application/json
Accept	application/json
Authorization	SKLMAuth userAuthId=<authIdValue>
Accept-Language	Any valid locale that is supported by IBM Security Key Lifecycle Manager. For example: en or de

Request body

JSON object with the following specification:

Parameter name	Description
name	Specify the alias of the key group to update. You must specify a value for either **name** or **uuid**. If both are specified, the values must match.

Request body

JSON object with the following specification:

Parameter name	Description
usage	Required. Specify a unique device group, such as `myNewLTO`, with the following values: **LTO** Specifies the `LTO` device group. **DS5000** Specifies the `DS5000` device group. **BRCD_ENCRYPTOR** Specifies the `BRCD_ENCRYPTOR` device group that is in the `LTO` device family. **ONESECURE** Specifies the `ONESECURE` device group that is in the `DS5000` device family. **GENERIC** Specifies a device family that uses the Key Management Interoperability Protocol to interact with IBM Security Key Lifecycle Manager. The `GENERIC` device group enables management of KMIP objects. Do not use REST service interface to add a device to the `GENERIC` device group or to change a `GENERIC` device group attribute. **ETERNUS_DX** Specifies the `ETERNUS_DX` device group that is in the `DS5000` device family. **XIV** Specifies the `XIV` device group that is in the `DS5000` device family. **userdevicegroup** Specifies a user-defined group that is based on a supported device family.
uuid	Specify the universal unique identifier of the key group. For example, `KEYGROUP-74386920-148c-47b2-a1e2-d19194b315cf`. You must specify a value for either **name** or **uuid**. If both are specified, the values must match.

Response

Response Headers

Header name	Value and description
Status Code	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.
Content-Type	`application/json`
Content-Language	Locale for the response message.

Success response body

JSON object with the following specification:

JSON property name	Description
code	Returns the code that is specified by the **status** property.
status	Returns the status to indicate whether the group update task was successful.

Error Response Body

JSON object with the following specification:

JSON property name	Description
code	Returns the application error code.
message	Returns a message that describes the error.

Examples

Service request to update a group

```
PUT https://localhost:9080/SKLM/rest/v1/keygroups
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
{"name":"myKeyGroup","usage":"myNewLTO"}
```

Success response

```
Status Code : 200 OK
{"code": "0","status": "Succeeded"}
```

Service request to update a group when group name is not specified

```
PUT https://localhost:9080/SKLM/rest/v1/keygroups
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
{"name": "", "usage":"myNewLTO"}
```

Error response

```
Status Code : 400 Bad Request
{"code": "CTGKM0630E","status": "CTGKM0630E Validation error:
"Invalid name " for parameter "name"."}
```

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.

Operation
PUT

URL https://<*host*>:<*port*>/SKLM/rest/v1/keys/export

Request

Request Parameters

Parameter	Description
host	Specify the IP address or host name of the IBM Security Key Lifecycle Manager server.
port	Specify the port number on which the IBM Security Key Lifecycle Manager server listens for requests.

Request Headers

Header name	Value
Content-Type	application/json
Accept	application/json
Authorization	SKLMAuth userAuthId=<authIdValue>
Accept-Language	Any valid locale that is supported by IBM Security Key Lifecycle Manager. For example: en or de

Request body

JSON object with the following specification:

Property name	Description
alias	Specifies an alias of the key that you export. This parameter is required if a value is not specified for the **aliasRange** parameter. For a privatekey type, a value for **alias** is required. For a secretkey type, you must specify a value for either **alias** or **aliasRange**.
aliasRange	This parameter is required if a value is not specified for the **alias** parameter. When the value of alias is specified, the value of **aliasRange** is ignored. To export a secret key, specify a three character prefix followed by a range of numbers in hexadecimal format. You can use the characters 0 through 9 and a through f. You can specify the range only for secret keys.
fileName	Specifies the relative or full path and the name of a file that IBM Security Key Lifecycle Manager creates to store the exported keys. If you do not specify a path name, the value of *SKLM_HOME* directory is used.

Request body

JSON object with the following specification:

Property name	Description
keyAlias	This parameter is required if the exported key is a secret key. Specify the alias of the public key entry in the keystore that is used to encrypt the secret key or keys to the file. Only the holder of the corresponding private key can access the keys.
password	This parameter is required if the value of the **type** parameter is privatekey. Specify a password to protect the PKCS#12 file to which the private key and certificate are exported. You might need to retain the value of the password to import the key.
type	Specifies whether the keys are secret or private. **secretkey** Specifies a symmetric key. **privatekey** Specifies an asymmetric key in a key pair with a public key and a private key. If you select this value, a password is required. If you export private keys to a PKCS#12 file, ensure that the file with the key is wrapped; use a FIPS-approved method before the file leaves the computer.

Response

Response Headers

Header name	Value and description
Status Code	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.
Content-Type	application/json
Content-Language	Locale for the response message.

Success response body

JSON object with the following specification:

JSON property name	Description
status	Returns the status to indicate whether the key is exported with an appropriate message.

Error Response Body

JSON object with the following specification:

JSON property name	Description
code	Returns the application error code.
message	Returns a message that describes the error.

Examples

Service request to export a private key

```
PUT https://localhost:9080/SKLM/rest/v1/keys/export
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
{"alias":"sklmCertificate","fileName":"myprivatekeys","type":
"privatekey",
"password":"mypassword"}
```

Success response

```
Status Code : 200 OK
{"code":"0","status":"Succeeded"}
```

Service request to export a secret key

```
PUT https://localhost:9080/SKLM/rest/v1/keys/export
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
{"aliasRange":"def0-3","fileName":"mysecretkeys","type":"secretkey",
"keyAlias":"sklmCertificate"}
```

Success Response

```
Status Code : 200 OK
{"status":"Exported Successfully"}
```

Error response

```
 Status Code : 400 Bad Request
{"code":"CTGKM0742E","message":"CTGKM0742E Key type not valid:
key."}
```

Key Group Default Rollover Add REST Service

Use **Key Group Default Rollover Add REST Service** to add a default key group rollover to serve keys to a device group on a specific date. The rollover key group takes the place of the previous default key group.

Operation
POST

URL https://*<host>*:*<port>*/SKLM/rest/v1/keygroups/rollover

Request

Request Parameters

Parameter	Description
host	Specify the IP address or host name of the IBM Security Key Lifecycle Manager server.
port	Specify the port number on which the IBM Security Key Lifecycle Manager server listens for requests.

Request Headers

Header name	Value
Content-Type	application/json
Accept	application/json
Authorization	SKLMAuth userAuthId=<authIdValue>
Accept-Language	Any valid locale that is supported by IBM Security Key Lifecycle Manager. For example: en or de

Request body

JSON object with the following specification:

JSON property name	Description
keyGroupName	Required. Specify the case-sensitive name of an existing key group.
usage	Required. Specify the device group. You can include the following values: **LTO** Specifies the LTO device group. The key is used in secure communication with LTO tape drives. **BRCD_ENCRYPTOR** Specifies the BRCD_ENCRYPTOR device group that is in the LTO device family. **userdevicegroup** Specifies a new, user-defined instance of the LTO device family. The value cannot exceed 16 characters in length. For example: myLTO.
effectiveDate	Required. Specify the rollover date on which the certificate becomes the default system or partner certificate. The value is a current or future date in yyyy-MM-dd format.

Response

Response Headers

Header name	Value and description
Status Code	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.
Content-Type	application/json
Content-Language	Locale for the response message.

Success response body

JSON object with the following specification:

JSON property name	Description
code	Returns the code that is specified by the **status** property.
status	Returns the status to indicate whether the key group is marked for rollover.

Error Response Body

JSON object with the following specification:

JSON property name	Description
code	Returns the application error code.
message	Returns a message that describes the error.

Examples

Service request to add a key group for rollover

```
POST https://localhost:9080/SKLM/rest/v1/keygroups/rollover
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
{"keyGroupName":"myLTOKeyGroup","usage":"LTO","effectiveDate":"2017-05-30"}
```

Success response

```
Status Code: 200 OK
{"code": "0","status": "Succeeded"}
```

Service request to add a key group for rollover with wrong usage

```
POST https://localhost:9080/SKLM/rest/v1/keygroups/rollover
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
{"keyGroupName":"myLTOKeyGroup","usage":"LTT","effectiveDate":"2017-05-30"}
```

Error response

```
Status Code: 400 Bad Request
{"code":"CTGKM0830E","message":"Device group is not valid: LTT"}
```

Key Group Default Rollover List REST Service

Use **Key Group Default Rollover List REST Service** to list key group rollovers in a rollover list.

Operation
GET

URL https://*<host>*:*<port>*/SKLM/rest/v1/keygroups/rollover?name=<name value>&usage<usage value>&uuid=<uuid value>

Request

Request Parameters

Parameter	Description
host	Specify the IP address or host name of the IBM Security Key Lifecycle Manager server.

Request Parameters

Parameter	Description
port	Specify the port number on which the IBM Security Key Lifecycle Manager server listens for requests.

Request Headers

Header name	Value
Content-Type	application/json
Accept	application/json
Authorization	SKLMAuth userAuthId=<authIdValue>
Accept-Language	Any valid locale that is supported by IBM Security Key Lifecycle Manager. For example: en or de

Query parameters

JSON property name	Description
name	Optional. Specify the case-sensitive name of a key group. For example: myLTOkeygroup
usage	Required. Specify the device group. You can include the following values: **LTO** Specifies the LTO device group. The key is used in secure communication with LTO tape drives. **BRCD_ENCRYPTOR** Specifies the BRCD_ENCRYPTOR device group that is in the LTO device family. **userdevicegroup** Specifies a new, user-defined instance of the LTO device family. The value cannot exceed 16 characters in length. For example: myLTO.
uuid	Optional. Specify the unique universal identifier of an existing key group rollover.

Response

Response Headers

Header name	Value and description
Status Code	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.

Response Headers

Header name	Value and description
Content-Type	application/json
Content-Language	Locale for the response message.

Success response body

JSON array that contains JSON objects with the following specification:

JSON property name	Description
Key group rollover uuid	Returns the unique universal identifier of the key group rollover.
<deviceGroup> system default	Returns the system default key group name for the device group.
Effective date	Returns the rollover date on which the key group becomes the default system key group. The value is a current or future date.

Error Response Body

JSON object with the following specification:

JSON property name	Description
code	Returns the application error code.
message	Returns a message that describes the error.

Examples

Service request to list key group rollover

```
GET https://localhost:9080/SKLM/rest/v1/keygroups/rollover?usage=LTO
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
```

Success response

```
 Status Code : 200 OK
[
 {
   "Key group rollover uuid":"1234",
   "LTO system default":"LTOSysDefKeyGroup",
   "Effective date":"2017-05-30 12:00:00 AM GMT-12:00"
 }
]
```

Service request to list key group rollover when an incorrect usage is specified

```
GET https://localhost:9080/SKLM/rest/v1/keygroups/rollover?usage=LTT
Content-Type: application/json
Accept : application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
```

Error response

```
Status Code : 400 Bad Request
{"code":"CTGKM0830E","message":"Device group is not valid: LTT"}
```

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.

To import secret keys, the import file might contain multiple keys. Each key contains the required alias value for that key. The import file must be generated by a previous **Key Export REST Service**.

Operation
POST

URL https://*<host>*:*<port>*/SKLM/rest/v1/keys/import

Request

Request Parameters

Parameter	Description
host	Specify the IP address or host name of the IBM Security Key Lifecycle Manager server.
port	Specify the port number on which the IBM Security Key Lifecycle Manager server listens for requests.

Request Headers

Header name	Value
Content-Type	application/json
Accept	application/json
Authorization	SKLMAuth userAuthId=<authIdValue>
Accept-Language	Any valid locale that is supported by IBM Security Key Lifecycle Manager. For example: en or de

Request body

JSON object with the following specification:

Property name	Description
alias	Required parameter if the value of the **type** attribute is secretkey and you want to rename the key with the **newAlias** parameter during the import process. Specify the value of alias if you want to import only this secret key from a secret key file that has other secret keys that you do not want to import. A value for alias is not required to import a private key because there is only one private key in the file. If you specify this value when you import a private key, the value is ignored.
fileName	Required. Specify the path and file name of the file from which the keys are imported.
keyAlias	This parameter is required if the value of the type attribute is secretkey. Specify the alias of the private key entry in the keystore that decrypts the secret key or keys, from the file. Use the same alias value to import and export a secret key or keys.
newAlias	Specify a new value for the key alias.

Request body

JSON object with the following specification:

Property name	Description
password	This parameter is required if the **type** parameter is privatekey. This password was previously specified with the **Key Export REST Service**. If you export private keys to a PKCS#12 file, ensure that the file with the key is wrapped with a FIPS-approved method before the file leaves the computer.
type	Specify whether the keys are secret or private. **secretkey** Specifies a symmetric key. If you select this value, specify a value for the **usage** attribute for a device group family that administers keys. **privatekey** Specifies an asymmetric key in a key pair with a public key and a private key. If you select this value, specify a value for the **usage** attribute for a device group that administers keys. You can specify any of the following values: • SSLCLIENT • SSLSERVER

Request body

JSON object with the following specification:

Property name	Description
usage	Specify the target application usage such as LTO device group. You can specify the following values: **LTO** Specifies the LTO device group. **3592** Specifies the 3592 device group. **DS5000** Specifies the DS5000 device group. **DS8000** Specifies the DS8000 device group. **BRCD_ENCRYPTOR** Specifies the BRCD_ENCRYPTOR device group that is in the LTO device family. **ONESECURE** Specifies the ONESECURE device group that is in the DS5000 device family. **GENERIC** Specifies a device family that uses the Key Management Interoperability Protocol to interact with IBM Security Key Lifecycle Manager. The GENERIC device group enables management of KMIP objects Do not use the REST interface to add a device to the GENERIC device group, or to change a GENERIC device group attribute. **SSLCLIENT** Client-side certificate that is used in secure communication by using Secure Socket Layer protocol to authenticate the client device. **SSLSERVER** Server-side certificate that is used in secure communication by using Secure Socket Layer protocol. **ETERNUS_DX** Specifies the ETERNUS_DX device group that is in the DS5000 device family. **XIV** Specifies the XIV device group that is in the DS5000 device family. **userdevicegroup** Specifies a user-defined group that is based on a supported device family.

Response

Response Headers

Header name	Value and description
Status Code	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.
Content-Type	application/json
Content-Language	Locale for the response message.

Success response body for `privatekey` type

JSON object with the following specification:

JSON property name	Description
code	Returns an integer value such as 0 to indicate the key import status.
status	Returns the status to indicate that the key import task is succeeded.

Success response body for `secretkey` type

JSON array that contains JSON objects with the following specification:

JSON property name	Description
ImportedKeys	JSON array that contains JSON objects with a list of imported keys. If no keys are imported, an empty list is returned.
ExistingKeys	JSON array that contains JSON objects with a list of duplicate keys. If there are no duplicate keys, an empty list is returned.
FailedToImportKeys	JSON array that contains JSON objects with a list of failed keys. If there are no keys failed keys, an empty list is returned.

Error Response Body

JSON object with the following specification:

JSON property name	Description
code	Returns the application error code.
message	Returns a message that describes the error.

Examples

Service request to import a symmetric key (secretkey type)

```
POST https://localhost:9080/SKLM/rest/v1/keys/import
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
{"keyAlias":" sklmCertificate", "alias":"xyz000000000000000000",
"newAlias":"ayz000000000000000000","type":"secretkey","fileName":
"myltokey00","usage":"LTO"}
```

Success response

```
 Status Code : 200 OK
{"ImportedKeys":[{"KeyAlias":"ayz000000000000000000"}],
"ExistingKeys":[],"FailedToImportKeys":[]}
```

Service request to import a private key (privatekey type)

```
POST https://localhost:9080/SKLM/rest/v1/keys/import
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
{"type":"privatekey","fileName":"mykey","usage":"SSLSERVER","password":
"mypassword","newAlias":"mykey"}
```

Success response

```
Status Code : 200 OK
{"code":"0","status":"Succeeded"}
```

Error response for an invalid request

```
POST https://localhost:9080/SKLM/rest/v1/keys/import
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
{"keyAlias":" sklmCertificate", "alias":"xyz000000000000000000",
"newAlias":"ayz000000000000000000","type":"secretkey","filename":
"myltokey01","usage":"LTO"}
```

Error response

```
Status Code : 500 Internal Server Error
{"code":"CTGKM0415E","CTGKM0415E Cannot find the file
C:\Program Files (x86)\IBM\WebSphere\AppServer\products\sklm\
myltokey01"}
```

Key Update REST Service

Use **Key Update REST Service** to update key metadata in the database. For example, you might move an individual key in one key group to another key group.

Operation
`PUT`

URL `https://<host>:<port>/SKLM/rest/v1/keys`

Request

Request Parameters

Parameter	Description
host	Specify the IP address or host name of the IBM Security Key Lifecycle Manager server.
port	Specify the port number on which the IBM Security Key Lifecycle Manager server listens for requests.

Request Headers

Header name	Value
Content-Type	application/json
Accept	application/json
Authorization	SKLMAuth userAuthId=<authIdValue>
Accept-Language	Any valid locale that is supported by IBM Security Key Lifecycle Manager. For example: en or de

Request body

Property name	Description
uuid	Specify the universal unique identifier of the individual key that you want to update.
usage	Optional. Specify a unique device group, such as LTO. You can specify the following values: **LTO** Specifies the LTO device group. **3592** Specifies the 3592 device group. **DS5000** Specifies the DS5000 device group. **DS8000** Specifies the DS8000 device group. **BRCD_ENCRYPTOR** Specifies the BRCD_ENCRYPTOR device group that is in the LTO device family. **ONESECURE** Specifies the ONESECURE device group that is in the DS5000 device family **GENERIC** Specifies a device family that uses the Key Management Interoperability Protocol to interact with Security Key Lifecycle Manager. The GENERIC device group enables management of KMIP objects. **ETERNUS_DX** Specifies the ETERNUS_DX device group that is in the DS5000 device family. **XIV** Specifies the XIV device group that is in the DS5000 device family. **userdevicegroup** Specifies a user-defined group that is based on a supported device family.

Request body

Property name	Description
attributes	Specify one or more of the following attribute-value pairs: **compromised** Specifies whether the use of a key is compromised. The only value is y (compromised). You cannot change a compromised key or certificate to an uncompromised state. **groupName** Specifies the name of a valid key group. You cannot move the last key in a default key group to a different group. You can change the key group name to a key group that is used by a different device group in the same device family if: • The key group and its keys are not the default device group. • The key group and its keys are not attached to a device. For example, you can change such a group from the myLTO device group to yourLTO device group in the LTO device family. In the DS5000 device family, a key group is generated for each DS5000 device when the device is created. You cannot create a DS5000 device with a key group attribute. However, you can create a new key group and specify the group name of a DS5000 device with the new key group. **information** *informationstring* Specifies more information about the use of an object.

Response

Response Headers

Header name	Value and description
Status Code	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.
Content-Type	application/json
Content-Language	Locale for the response message.

Success response body

JSON object with the following specification:

JSON property name	Description
status	Returns the status with an appropriate message that indicates whether the key is updated.

Error Response Body

JSON object with the following specification:

JSON property name	Description
code	Returns the application error code.
message	Returns a message that describes the error.

Examples

Service request to update the group and key details

```
PUT https://<host>:<port>/SKLM/rest/v1/keys
Content-Type: application/json
Accept : application/json
Authorization : SKLMAuth userAuthId=37ea1939-1374-4db7-84cd-14e399be2d20
Accept-Language : en
{"uuid":"KEY-61bd4100-9880-450f-a1a5-7efe19a8d0f5","attributes":"groupName
newGroup1,information movedTonewGroup1"}
```

Success response

```
Status Code : 200 OK
Content-Language: en
{"code":"0","status":"Succeeded"}
```

Service request to update the key when `uuid` parameter is missing

```
PUT https://<host>:<port>/SKLM/rest/v1/keys
Content-Type: application/json
Accept : application/json
Authorization : SKLMAuth userAuthId=37ea1939-1374-4db7-84cd-14e399be2d20
Accept-Language : en
{}
```

Error response

```
Status Code: 400 Bad Request
Content-Language: en
{ "code": "CTGKM0631E", "message": "CTGKM0631E Missing required
parameter " uuid " ."}
```

Last Backup Info REST Service

Use **`Last Backup Info REST Service`** to request for the last backup information of IBM Security Key Lifecycle Manager data.

Operation
GET

URL `http://<host>:<port>/SKLM/rest/v1/backupInfo`

Request

Request Parameters

Parameter	Description
host	Specify the IP address or host name of the IBM Security Key Lifecycle Manager server.
port	Specify the port number on which the IBM Security Key Lifecycle Manager server listens for requests.

Request Headers

Header name	Value
Content-Type	application/json
Accept	application/json
Authorization	SKLMAuth userAuthId=<authIdValue>
Accept-Language	Any valid locale that is supported by IBM Security Key Lifecycle Manager. For example: en or de

Response

Response Headers

Header name	Value and description
Status Code	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.
Content-Type	application/json
Content-Language	Locale for the response message.

Success response body

JSON Object with the following specification:

JSON property name	Description
IsBackupBeforeServing	Indicates whether the keys are released after the backup operation.
IsLastBackupDateAvailable	Indicates whether the last backup date for IBM Security Key Lifecycle Manager data is available.
lastBackupDate	Returns the last backup date of IBM Security Key Lifecycle Manager data.

Error Response Body

JSON object with the following specification:

JSON property name	Description
code	Returns the application error code.
message	Returns a message that describes the error.

Examples

Service request to get backup information

```
GET http://<host>:<port>/SKLM/rest/v1/backupInfo
Content-Type: application/json
Accept : application/json
Authorization : SKLMAuth userAuthId=37ea1939-1374-4db7-84cd-14e399be2d20
Accept-Language : en
```

Success response

```
Status Code : 200 OK
Content-Language: en
{"isBackupBeforeServing" : "true",
"isLastBackupDate" : "true",
"lastBackupDate" : "27497358095"}
```

Backup does not exist and the **isBackupBeforeServing** property is set to “false”

```
Status Code : 200 OK
Content-Language: en
{"isBackupBeforeServing" : "false",
"isLastBackupDate" : "false"}
```

Error response

```
Status Code : 400 Bad Request
Content-Language: en
{"code" : "CTGKM6002E", "message" : "message":"CTGKM6002E Bad Request:
Invalid user authentication ID or invalid request format."}
```

List Key REST Service

Use **List Key REST Service** to list a key or keys in the keystore that is based on specified criteria, such as an active state.

List Key REST Service supports pagination. The request parameters, such as offset and count, are used for pagination. For example, to retrieve the first 10 records for the list, set **offset = 1** and **count = 10**. To retrieve the next 10 records, set **offset = 2** and **count = 10**. If you do not specify values for pagination parameters, the first 2000 records are returned.

Operation

GET

URL

To retrieve all keys:

https://*<host>*:*<port>*/SKLM/rest/v1/keys

Note: Returns the first 2000 records.

To retrieve a specific list:

https://*<host>*:*<port>*/SKLM/rest/v1/keys?uuid=<uuid>
&alias=<alias>&usage=<usage>&attributes=<state value>

Note: Returns the first 2000 records.

To retrieve a specific list with pagination:

```
https://<host>:<port>/SKLM/rest/v1/keys?uuid=<uuid>
&alias=<alias>&attributes=<state value>&usage=<usage>
&offset=<offset>&count=<count>
```

Request

Request Parameters

Parameter	Description
host	Specify the IP address or host name of the IBM Security Key Lifecycle Manager server.
port	Specify the port number on which the IBM Security Key Lifecycle Manager server listens for requests.

Request Headers

Header name	Value
Content-Type	application/json
Accept	application/json
Authorization	SKLMAuth userAuthId=<authIdValue>
Accept-Language	Any valid locale that is supported by IBM Security Key Lifecycle Manager. For example: en or de

Query parameters

Parameter name	Description
uuid	Specify the unique ID of the key.
alias	Specify the unique alias name for the key.
usage	Specify the device group that contains the key. You can specify the following values: • LTO • 3592 • DS5000 • DS8000 • BRCD • ENCRYPTOR • ONESECURE • GENERIC • ETERNUS_DX • XIV • userdevicegroup • SSLSERVER • SSLCLIENT

Query parameters

Parameter name	Description
state	Specify the current state of the key. You can specify the following values: • pending • pre-active • active • compromised • deactivated • destroyed • destroyed-compromised
offset	Specify the page number from which the records are displayed based on the value that you specify for **count**.
count	Specify the number of records to display on the specified page that you specify with **offset**.

Response

Response Headers

Header name	Value and description
Status Code	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.
Content-Type	application/json
Content-Language	Locale for the response message.

Success response body

JSON object with the following specification:

JSON property name	Description
uuid	Returns the unique ID of the key.
alias	Returns the alias name for the key.
information	Returns the important information about the key.
key algorithm	Represents the algorithm to encrypt the key.
Key length	Returns the length of the key in bits.
Key type	Indicates key type, such as symmetric.
key store name	Returns the keystore name that contains the key.
key store uuid	Returns the unique ID of the keystore that contains the key.

Success response body

JSON object with the following specification:

JSON property name	Description
owner	Returns the key owner name.
key state	Returns the status of the key, such as ACTIVE.
activation date	Returns the key activation date.
archive date	Returns the Key archived date.
compromise date	Returns the date on which the key is compromised.
creation date	Returns the key creation date.
expiration date	Returns the key expiration date.
destroy date	Returns the date on which the key is destroyed.
Key group id	Returns the unique ID of the group that contains the key.
hash value	Returns the hash value of the key.
usage	Indicates the usage type of the target application such as LTO, 3592, or DS5000.
Deactivation Date	Identifies the date on which the key is deactivated.
Cryptographic Usage Mask	Indicates the cryptographic usage of a key. For example, Encrypt, Decrypt, or Export.
Operation Policy Name	Identifies operation policy that controls the key management operations on the cryptographic object.
Contact Information	Represents the contact information.
Revocation Reason	Indicates the reason for revoking the managed cryptographic. For example, compromised, expired, or no longer used
Name	Returns the name that identifies and locates the cryptographic object.
Cryptographic Parameters	Returns the parameters for cryptographic operations.
Object Group	Returns the group that contains the cryptographic object.
Link	Identifies the target managed cryptographic object by its unique identifier.
Digest	Contains the digest value of the key or secret data.
Application Specific Information	Stores information specific to the application that uses managed cryptographic object.
Custom Attributes	Returns the vendor defined custom attributes.
Last Changed Date	Returns the date and time of the last change to the contents or attributes of the specified managed cryptographic object.
Compromise Occurrence Date	Returns the date and time of when the managed cryptographic object was first believed to be compromised.
Lease Time	Defines a time interval for a managed cryptographic object. After the lease time, the client cannot use the object without obtaining another lease.

Error Response Body

JSON object with the following specification:

JSON property name	Description
`code`	Returns the application error code.
`message`	Returns a message that describes the error.

Examples

Service request to list key information

```
GET https://localhost:9080/SKLM/rest/v1/keys?offset=1&count=1
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
```

Success response

```
Status Code: 200 OK
    [
       {
           "uuid": "KEY-d58c43dc-302a-42e0-8c7a-2543a7d19548",
           "information": "",
           "alias": "aaa000000000000000000 ",
           "key algorithm": "AES",
           "key length (in bits)": "256",
           "key type": "SYMMETRIC_KEY",
           "owner": "",
           "key store name": "defaultKeyStore ",
           "key store uuid": "KEYSTORE-edff2086-5eaa-4bf0-b2e8-
             05964ad8eeea ",
           "key state": "ACTIVE",
           "activation date": "12/20/12 1:05:45 PM Central Standard
             Time",
           "archive date": "null",
           "compromise date": "null",
           "creation date": "12/20/12 1:05:45 PM Central Standard
             Time",
           "expiration date": "12/15/32 1:05:45 PM Central Standard
             Time",
           "destroy date": "null",
           "key group ids": "KEYGROUP-b0992af0-ae98-4a89-9c3c-
             dcc30ad6a347 ",
           "hash value": "0000: 02 49 e1 23 4d 66 20 bc ee 33 5d d0
             e8 ff af de .I..Mf...3...... 0010: 8f 0c ac 28 07 cf
             a5 99 57 7d 18 1f 51 aa 15 5b ........W...Q... ",
           "usage": "LTO",
           "Cryptographic Usage Mask": "",
           "Usage Limits": "",
           "Operation Policy Name": "",
           "Process Start Date": "",
           "Protect Stop Date": "",
           "Deactivation Date": "12/15/32 1:05:45 PM Central
             Standard Time",
           "Contact Information": "",
           "Revocation Reason": "",
           "Name": "[[INDEX 0] [TYPE TEXT] [VALUE aaa0000000000000
             00000]]",
           "Cryptographic Parameters": "",
           "Object Group": "",
           "Link": "",
           "Digest": "[[INDEX 0] [HASH SHA256] [VALUE x02,x49,xe1,
             x23,x4d,x66,x20,xbc,xee,x33,x5d,xd0,xe8,xff,xaf,xde,
             x8f,x0c,xac,x28,x07,xcf,xa5,x99,x57,x7d,x18,x1f,x51,
             xaa,x15,x5b]]",
           "Application Specific Information": "",
                "Custom Attributes": "",
                "Last Changed Date": "12/20/12 1:05:45 PM Central
                  Standard Time",
                "Compromise Occurence Date": "",
                "Lease Time": ""
            }
        ]
```

Error response

```
Status Code: 400 Bad Request
Content-Language: en

{"code" "CTGKM6002E"
 , "message": "CTGKM6002E Bad Request: Invalid user authentication
ID or invalid re-quest format."
 }
```

Machine Device Add REST Service

Use **Machine Device Add REST Service** to add the association of a device to an existing machine identifier in the IBM Security Key Lifecycle Manager database.

Operation
`POST`

URL `https://<host>:<port>/SKLM/rest/v1/machines/device`

Request

Request Parameters

Parameter	Description
host	Specify the IP address or host name of the IBM Security Key Lifecycle Manager server.
port	Specify the port number on which the IBM Security Key Lifecycle Manager server listens for requests.

Request Headers

Header name	Value
Content-Type	application/json
Accept	application/json
Authorization	SKLMAuth userAuthId=<authIdValue>
Accept-Language	Any valid locale that is supported by IBM Security Key Lifecycle Manager. For example: en or de

Request body

JSON object with the following specification:

Property name	Description
deviceUUID	Required. Specify a value for a unique universal identifier for the device, such as DEVICE-7d588437-e725-48bf-a836-00a47df64e78. Use **Device List REST Service** to locate the device uuid.
machineID	Required if you do not specify the value of **machineText**. Specify a unique ID in a range 1 - 48 characters. For example: 30423830303034370000000000000. Use **Machine Identity List REST Service** to locate machine identities.

Request body

JSON object with the following specification:

Property name	Description
machineText	Required if you do not specify the value of machineID. Specify a unique, user-supplied descriptive label greater than zero bytes and not more than 96 bytes for a machine, such as myEncryptedDS5000.

Response

Response Headers

Header name	Value and description
Status Code	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.
Content-Type	application/json
Content-Language	Locale for the response message.

Success response body

JSON object with the following specification:

JSON property name	Description
code	Returns the code that is specified by the status property.
status	Returns the status to indicate whether the addition of a machine device association was successful.

Error Response Body

JSON object with the following specification:

JSON property name	Description
code	Returns the application error code.
message	Returns a message that describes the error.

Examples

Service request to add a machine device association when device and machine exist

```
POST https://localhost:9080/SKLM/rest/v1/machines/device
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
{"deviceUUID":"DEVICE-7d588437-e725-48bf-a836-00a47df64e78","machineID":
"3042383030303437000000000000"}
```

Success response

```
Status Code: 200 OK
{"code":"0","status":"Succeeded"}
```

Error response when there is no device

```
POST https://localhost:9080/SKLM/rest/v1/machines/device
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
{"deviceUUID":"DEVICE-7d588437-e725-48bf-a836-00a47df64e78","machineID":
"3042383030303437000000000000"}
```

Error response

```
 Status Code: 500 Internal Server Error
{"code":"CTGKM1406E","message":"CTGKM1406E Device does not exist."}
```

Machine Device Delete REST Service

Use **`Machine Device Delete REST Service`** to remove the association of a device to an existing machine identifier in the IBM Security Key Lifecycle Manager database.

Operation
`DELETE`

URL `https://<host>:<port>/SKLM/rest/v1/machines/device`

Request

Request Parameters

Parameter	Description
host	Specify the IP address or host name of the IBM Security Key Lifecycle Manager server.
port	Specify the port number on which the IBM Security Key Lifecycle Manager server listens for requests.

Request Headers

Header name	Value
Content-Type	application/json
Accept	application/json
Authorization	SKLMAuth userAuthId=<authIdValue>
Accept-Language	Any valid locale that is supported by IBM Security Key Lifecycle Manager. For example: en or de

Request body

JSON object with the following specification:

Property name	Description
deviceUUID	Required. Specify a value for a unique universal identifier for the device, such as DEVICE-7d588437-e725-48bf-a836-00a47df64e78. Use **Device List REST Service** to locate the device uuid.
machineID	Required parameter if you do not specify the value of **machineText**. Specify a unique ID in a range 1 - 48 characters. For example: 30423830303034370000000000000. Use **Machine Identity List REST Service** to locate machine identities.
machineText	Required parameter if you do not specify the value of **machineID** or **deviceUUID**. Specify a unique, user-supplied descriptive label greater than zero bytes and not more than 96 bytes for a machine, such as myEncryptedDS5000.

Response

Response Headers

Header name	Value and description
Status Code	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.
Content-Type	application/json
Content-Language	Locale for the response message.

Success response body

JSON object with the following specification:

JSON property name	Description
code	Returns the code that is specified by the status property.
status	Returns the status to indicate whether the removal of a machine device association was successful.

Error Response Body

JSON object with the following specification:

JSON property name	Description
code	Returns the application error code.

Error Response Body

JSON object with the following specification:

JSON property name	Description
message	Returns a message that describes the error.

Examples

Service request to remove a machine device association when device and machine exist

```
DELETE https://localhost:9080/SKLM/rest/v1/machines/device
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
{"deviceUUID":"DEVICE-7d588437-e725-48bf-a836-00a47df64e78","machineID":
"3042383030303437000000000000"}
```

Success response

```
 Status Code : 200 OK
{"code":"0","status":"Succeeded"}
```

Error response when there is no device association

```
DELETE https://localhost:9080/SKLM/rest/v1/machines/device
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
{"deviceUUID":"DEVICE-7d588437-e725-48bf-a836-00a47df64e78","machineID":
"3042383030303437000000000000"}
```

Error response

```
Status Code : 500 Internal Server Error
{"code":"CTGKM1436E","message":"CTGKM1436E No machine affinity
between device DEVICE-7d588437-e725-48bf-a836-00a47df64e78 and
machine 3042383030303437000000000000."}
```

Machine Device List REST Service

Use **Machine Device List REST Service** to list all the devices that are associated with a specific machine ID or machine text.

Machine Device List REST Service supports pagination. The request parameters, such as offset and count, are used for pagination. For example, to retrieve the first 10 records for the list, set **offset = 1** and **count = 10**. To retrieve the next 10 records, set **offset = 2** and **count = 10**. If you do not specify values for pagination parameters, the first 2000 records are returned.

Operation

GET

URL

To retrieve all devices:

https://*<host>*:*<port>*/SKLM/rest/v1/machines/device?machineID=<machineID>&machineText=<machineText>

Note: Returns the first 2000 records.

To retrieve a specific list with pagination:

https://*<host>*:*<port>*/SKLM/rest/v1/machines/device?machineID=<machineID>&machineText=<machineText>&offset=<offset>&count=<count>

Request

Request Parameters

Parameter	Description
host	Specify the IP address or host name of the IBM Security Key Lifecycle Manager server.
port	Specify the port number on which the IBM Security Key Lifecycle Manager server listens for requests.

Request Headers

Header name	Value
Content-Type	application/json
Accept	application/json
Authorization	SKLMAuth userAuthId=<authIdValue>
Accept-Language	Any valid locale that is supported by IBM Security Key Lifecycle Manager. For example: en or de

Query parameters

Parameter name	Description
machineID	Required if the value of **machineText** is not specified. Specify a unique machine ID in a range 1 - 48 characters. For example, 30423830303034370000000000000. Use **Machine Identity List REST Service** to locate machine identities.
machineText	Required if the value of **machineID** is not specified. Specify a unique, user-supplied descriptive label greater than zero bytes and not more than 96 bytes for a machine, such asmyEncryptedDS5000.
offset	Optional. Specify the page number from which the records are displayed based on the value that you specify for **count**.
count	Optional. Specify the number of records to display on the page that you specified with **offset**. The count must not exceed 2000 records.

Response

Response Headers

Header name	Value and description
Status Code	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.

Response Headers

Header name	Value and description
Content-Type	application/json
Content-Language	Locale for the response message.

Success response body

JSON object with the following specification:

JSON property name	Description
Description	Returns the additional information that describes the type of device or its purpose.
Serial Number	Returns the serial number of the device.
Device uuid	Returns the universal unique identifier of the device.
Device group	Returns the device group that contains the device.
World wide name	Returns the name of the device.
Sym alias	Returns the alias name that is used to identify an existing key group.

Error Response Body

JSON object with the following specification:

JSON property name	Description
code	Returns the application error code.
message	Returns a message that describes the error.

Examples

Service request to obtain a list for a machine ID with existing associations

```
GET https://localhost:9080/SKLM/rest/v1/machines/device?machineID=
3042383030303437000000000000
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
```

Success response

```
Status Code : 200 OK
    [
        {
     "Description":"marketingDivisionDrive01",
     "Serial Number" : "CDA39403DD1",
     "Device uuid" : "DEVICE-b8e73d20-9fa4-43f7-8669-99656e081d23",
     "Device group" : "DS5000",
     "World wide name" : "ABCDEF1234567445",
     "Sym alias" : "DS5K-CDA39403DD1"
    },
    {
     "Description" : "marketingDivisionDrive02",
     "Serial Number" : "CDA39403DDD",
     "Device uuid" : "DEVICE-e0a93b2f-cafc-4e30-9a86-215f823999e3",
     "Device group" : "DS5000",
     "World wide name" : "ABCDEF1234567444",
     "Sym alias" : "DS5K-CDA39403DDD"
    }
  ]
```

Service request to obtain a list when machineID not found

```
GET https://localhost:9080/SKLM/rest/v1/machines/device?machineID=
304238303030343700000000000
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
```

Error response

```
Status Code : 400 Bad Request
{"code":"CTGKM1408E ","message":"CTGKM1408E Machine ID/Text not
found."}
```

Machine Identity Add REST Service

Use **Machine Identity Add REST Service** to create a machine identity in the IBM Security Key Lifecycle Manager database.

Operation
POST

URL `https://<host>:<port>/SKLM/rest/v1/machines`

Request

Request Parameters

Parameter	Description
host	Specify the IP address or host name of the IBM Security Key Lifecycle Manager server.
port	Specify the port number on which the IBM Security Key Lifecycle Manager server listens for requests.

Request Headers

Header name	Value
Content-Type	application/json
Accept	application/json
Authorization	SKLMAuth userAuthId=<authIdValue>
Accept-Language	Any valid locale that is supported by IBM Security Key Lifecycle Manager. For example: en or de

Request body

JSON object with the following specification:

Property name	Description
machineID	Required. Specify a unique ID in a range 1 - 48 characters. For example: 304238303030343700000000000. Use **Machine Identity List REST Service** to locate machine identities.
machineText	Optional. Specify a unique, user-supplied descriptive label greater than zero bytes and not more than 96 bytes for a machine, such as myEncryptedDS5000.

Response

Response Headers

Header name	Value and description
Status Code	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.
Content-Type	application/json
Content-Language	Locale for the response message.

Success response body

JSON object with the following specification:

JSON property name	Description
code	Returns the code that is specified by the status property.
status	Returns the status to indicate whether the addition of a machine identity to the database was successful.

Error Response Body

JSON object with the following specification:

JSON property name	Description
code	Returns the application error code.
message	Returns a message that describes the error.

Examples

Service request to add a machine identity

```
POST https://localhost:9080/SKLM/rest/v1/machines
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
{"machineID":"30423830303034370000000000000"}
```

Success response

```
Status Code : 200 OK
{"code":"0","status":"Succeeded"}
```

Error response when a machine identity exists

```
POST https://localhost:9080/SKLM/rest/v1/machines
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
{"machineID":"30423830303034370000000000000"}
```

Error response

```
Status Code : 500 Internal Server Error
{"code":"CTGKM1426E","message":"CTGKM1426E Machine ID is not unique."}
```

Machine Identity Delete REST Service

Use **Machine Identity Delete REST Service** to remove a machine identity from the IBM Security Key Lifecycle Manager database.

Operation
DELETE

URL https://<*host*>:<*port*>/SKLM/rest/v1/machines

Request

Request Parameters

Parameter	Description
host	Specify the IP address or host name of the IBM Security Key Lifecycle Manager server.
port	Specify the port number on which the IBM Security Key Lifecycle Manager server listens for requests.

Request Headers

Header name	Value
Content-Type	application/json
Accept	application/json
Authorization	SKLMAuth userAuthId=<authIdValue>
Accept-Language	Any valid locale that is supported by IBM Security Key Lifecycle Manager. For example: en or de

Request body

JSON object with the following specification:

Property name	Description
machineID	Required if you do not specify the value of **machineText** or **machineUUID**. Specify a unique ID in a range 1 - 48 characters. For example: 30423830303034370000000000000. Use **Machine Identity List REST Service** to locate machine identities.
machineText	Required if you do not specify the value of **machineID** or **machineUUID**. Specify a unique, user-supplied descriptive label greater than zero bytes and not more than 96 bytes for a machine, such as myEncryptedDS5000.
machineUUID	Required if you do not specify the value of **machineText** or **machineID**. Specify a value for a unique universal identifier for the machine, such as MACHINE-bf57ca0d-1299-4bc7-9c9c-2fa29a99c7c9. Use **Machine List REST Service** to locate the machine uuid.

Response

Response Headers

Header name	Value and description
Status Code	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.
Content-Type	application/json
Content-Language	Locale for the response message.

Success response body

JSON object with the following specification:

JSON property name	Description
code	Returns the code that is specified by the status property.
status	Returns the status to indicate whether the removal of a machine identity from the database was successful.

Error Response Body

JSON object with the following specification:

JSON property name	Description
code	Returns the application error code.
message	Returns a message that describes the error.

Examples

Service request to remove a machine identity

```
DELETE https://localhost:9080/SKLM/rest/v1/machines
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
{machineID":"3042383030303437000000000000"}
```

Success response

```
Status Code: 200 OK
{"code":"0","status":"Succeeded"}
```

Error response when there is no machine identity

```
DELETE https://localhost:9080/SKLM/rest/v1/machines
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
{machineID":"3042383030303437000000000000"}
```

Error response

```
Status Code: 500 Internal Server Error
{"code":"CTGKM1408E","message":"CTGKM1408E Machine ID/Text not
found."}
```

Machine Identity List REST Service

Use **Machine Identity List REST Service** to list known machine identities for a DS5000 device group.

Machine Identity List REST Service supports pagination. The request parameters, such as offset and count, are used for pagination. For example, to retrieve the first 10 records for the list, set **offset = 1** and **count = 10**. To retrieve the next 10 records, set **offset = 2** and **count = 10**. If you do not specify values for pagination parameters, the first 2000 records are returned.

Operation

GET

URL

To retrieve all machine identities:

https://<*host*>:<*port*>/SKLM/rest/v1/machines

Note: Returns the first 2000 records.

To retrieve a specific list with pagination:

https://<*host*>:<*port*>/SKLM/rest/v1/machines?offset=<offset>&count=<count>

Request

Request Parameters

Parameter	Description
host	Specify the IP address or host name of the IBM Security Key Lifecycle Manager server.
port	Specify the port number on which the IBM Security Key Lifecycle Manager server listens for requests.

Request Headers

Header name	Value
Content-Type	application/json
Accept	application/json
Authorization	SKLMAuth userAuthId=<authIdValue>
Accept-Language	Any valid locale that is supported by IBM Security Key Lifecycle Manager. For example: en or de

Query parameters

Parameter name	Description
offset	Optional. Specify the page number from which the records are displayed based on the value that you specify for **count**.
count	Optional. Specify the number of records to display on the page that you specified with **offset**. The count must not exceed 2000 records.

Response

Response Headers

Header name	Value and description
Status Code	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.
Content-Type	application/json
Content-Language	Locale for the response message.

Success response body

JSON object with the following specification:

JSON property name	Description
Machine ID	Returns the unique machine identification.
Machine UUID	Returns the universal unique identifier of the machine.
Machine Text	Returns the machine text of the machine that helps you to uniquely identify the machine.
Device Group	Returns the device group of the machine.

Error Response Body

JSON object with the following specification:

JSON property name	Description
code	Returns the application error code.
message	Returns a message that describes the error.

Examples

Service request to list the machine identities

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

Success response

```
Status Code : 200 OK
[
   {
        "Machine ID":"30423830303034370000000005110",
    "Machine UUID" : "MACHINE-9efafff1-1df2-4cf9-abd4-fefee4d72508",
    "Device group" : "DS5000"
  },
   {
```

```
                 "Machine ID":"3042383030303437000000005111",
              "Machine UUID" : "MACHINE-f33140c8-a2bc-4aa0-afd3-f94a6208dbf0",
              "Machine Text" : "machineText001",
              "Device group" : "DS5000"
             }
          ]
```

Service request to list the machine identities when you specify incorrect parameter

```
GET https://localhost:9080/SKLM/rest/v1/machines/COUNT=2&offset=1
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
```

Error response

```
Status Code : 400 Bad Request
{"code":"CTGKM0630E","message":"CTGKM0630E Validation error:
\"Invalid name \" for parameter \"Count\"."}
```

Master Key REST Service

Use **Master Key REST Service** to create an IBM Security Key Lifecycle Manager master key of the length you specify for data encryption.

The master key that you created by using **Master Key REST Service** encrypts all the data that was encrypted with the old master key. Multiple reruns of **Master Key REST Service** are possible if any failure during a run of this REST service.

Operation
`POST`

URL `https://<host>:<port>/SKLM/rest/v1/ckms/masterKey`

Request

Request Parameters

Parameter	Description
host	Specify the IP address or host name of the IBM Security Key Lifecycle Manager server.
port	Specify the port number on which the IBM Security Key Lifecycle Manager server listens for requests.

Request Headers

Header name	Value
Content-Type	application/json
Accept	application/json
Authorization	SKLMAuth userAuthId=<authIdValue>
Accept-Language	Any valid locale that is supported by IBM Security Key Lifecycle Manager. For example: en or de

Request Body

JSON object with the following specification:

JSON property name	Description
masterKeySize	Specify length of the IBM Security Key Lifecycle Manager master key in bits. You can specify 128 or 256.

Response

Response Headers

Header name	Value and description
Status Code	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.
Content-Type	`application/json`
Content-Language	Locale for the response message.

Success Response Body

JSON object with the following specification:

JSON property name	Description
status	Returns the status to indicate whether the IBM Security Key Lifecycle Manager master key is created with an appropriate message.

Error Response Body

JSON object with the following specification:

JSON property name	Description
code	Returns the application error code.
message	Returns a message that describes the error.

Examples

Service request to create a new master key with 256-bits length

```
POST https://localhost:9080/SKLM/rest/v1/ckms/masterKey
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
Accept-Language: en
{"masterKeySize":"256"}
```

Success response

```
Status Code: 200 OK
{"status":"Succeeded"}
```

Error response

```
Status Code: 400 Bad Request
Content-Language: en
{"code": "CTGKM6002E", "message" : "CTGKM6002E Bad Request: Invalid user
authentication ID or invalid request format."}
```

Pending Client Certificate Accept REST Service

Use **Pending Client Certificate Accept REST Service** to accept the client communication certificate that was pushed to the IBM Security Key Lifecycle Manager server from a client device for secure communication. The certificate is added to the keystore and marked as trusted.

Operation
POST

URL https://*<host>*:*<port>*/SKLM/rest/v1/pendingClientCertificates

Request

Request Parameters

Parameter	Description
host	Specify the IP address or host name of the IBM Security Key Lifecycle Manager server.
port	Specify the port number on which the IBM Security Key Lifecycle Manager server listens for requests.

Request Headers

Header name	Value
Content-Type	application/json
Accept	application/json
Authorization	SKLMAuth userAuthId=<authIdValue>
Accept-Language	Any valid locale that is supported by IBM Security Key Lifecycle Manager. For example: en or de

Request body

JSON object with the following specification:

Property name	Description
alias	Specify a unique name for the certificate. The alias that is passed in the request, is the alias of the accepted certificate. The name is not case-sensitive. For example, if you specify MY Cert1, the value is stored as my cert1. Do not use: • The value that begins with 3 alphabetic characters followed by 18 numeric characters, such as aaa000000000000000002. IBM Security Key Lifecycle Manager uses this format to generate a key group with symmetric keys. • Forward slash (/) or backslash (\) characters in the value.
uuid	Specify the universal unique identifier of the certificate in the IBM Security Key Lifecycle Manager database, such as CERTIFICATE-4e064e39-5c15-4e29-83ab-ebd4d275e148.

Response

Response Headers

Header name	Value and description
Status Code	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.
Content-Type	`application/json`
Content-Language	Locale for the response message.

Success response body

JSON object with the following specification:

JSON property name	Description
code	Returns the code that is specified by the status property.
status	Returns the status to indicate whether the pending client certificate was successfully accepted.

Error Response Body

JSON object with the following specification:

JSON property name	Description
code	Returns the application error code.
message	Returns a message that describes the error.

Examples

Service request to accept a pending client certificate

```
POST https://localhost:9080/SKLM/rest/v1/pendingClientCertificates
Content-Type: application/json
Accept : application/json
Authorization : SKLMAuth userAuthId=37ea1939-1374-4db7-84cd-14e399be2d20
Accept-Language : en
{"alias":"sklmcertificate1","uuid":"CERTIFICATE-a38e2239-a36b-41ff-97a1-
0ca72cfa08e8"}
```

Success response

```
Status Code: 200 OK
{"code": "0","status": "Succeeded"}
```

Error response when certificate alias exists

```
Status Code : 500 Internal Server Error
{"code":"CTGKM2306E","message":"CTGKM2306E  Client certificate alias
already in use: sklmcertificate1 "}
```

Error response when there is no pending client certificate

```
Status Code : 500 Internal Server Error
{"code":"CTGKM2307E","message":"CTGKM2307E  Client certificate UUID
not found in the database: CERTIFICATE-a3862239-a367-41ff-97a1-
0ca72cfa08e8 "}
```

Pending Client Certificate List REST Service

Use **`Pending Client Certificate List REST Service`** to list pending certificates that are pushed to the server from a client device for secure communication with IBM Security Key Lifecycle Manager.

Acceptance marks the certificate as trusted and allows the client device to establish secure communication with IBM Security Key Lifecycle Manager. The certificate is added to the keystore. Rejection removes the certificate from the pending list and prevents its use for secure communication between the device and the server.

Operation
`GET`

URL `https://<host>:<port>/SKLM/rest/v1/pendingClientCertificates`

Request

Request Parameters

Parameter	Description
host	Specify the IP address or host name of the IBM Security Key Lifecycle Manager server.
port	Specify the port number on which the IBM Security Key Lifecycle Manager server listens for requests.

Request Headers

Header name	Value
Content-Type	application/json
Accept	application/json
Authorization	SKLMAuth userAuthId=<authIdValue>
Accept-Language	Any valid locale that is supported by IBM Security Key Lifecycle Manager. For example: en or de

Response

Response Headers

Header name	Value and description
Status Code	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.
Content-Type	application/json
Content-Language	Locale for the response message.

Success response body

JSON object with the following specification:

JSON property name	Description
uuid	Returns the universal unique identifier of the certificate.
subject name	Returns the certificate subject name. The X.509 certificates contain the subject distinguished name. The property value is from the **Subject** field of the certificate.
issuer name	Returns the distinguished name of the certificate issuer. The property value is from the **Issuer** field of the certificate.
serial number	Returns the certificate serial number.
client cert pending date	Returns the certificate pending date for acceptance or rejection.
key state	Indicates the certificate status, such as ACTIVE.
creation date	Returns the certificate creation date.
expiration date	Returns the certificate expiration date at which the certificate expires for use in secure communication.
Cryptographic Algorithm	Represents the cryptographic algorithm of the certificate, such as RSA, DSA, DES, 3DES, or AES.
Cryptographic Length	Returns the length of the clear-text cryptographic object in bits.
X509 output	Returns the standard X509 certificate details.

Error Response Body

JSON object with the following specification:

JSON property name	Description
code	Returns the application error code.
message	Returns a message that describes the error.

Examples

Service request to list pending client certificates

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

Success response

```
Status Code : 200 OK
[
  {
    "uuid": "CERTIFICATE-a2b3f39e-e2e2-4c3a-8d2a-1a7a70325f98",
    "subject name": "CN=sklm, OU=sales, O=myCompanyName, C=US",
    "issuer name": "CN=sklm, OU=sales, O=myCompanyName, C=US",
    "serial number": "187046468526998",
    "client cert pending date": "null",
    "key state": "ACTIVE",
    "creation date": "2/5/14 2:36:08 PM India Standard Time",
    "expiration date": "10/31/16 2:36:08 PM India Standard Time",
    "Cryptographic Algorithm": "null",
    "Cryptographic Length": "null",
    "X509 output": "[
[
  Version: V3
  Subject: CN=sklm, OU=sales, O=myCompanyName, C=US
  Signature Algorithm: SHA256withRSA, OID = 1.2.840.113549.1.1.11

  Key:  IBMJCE RSA Public Key:
modulus:
20343792752701333201377325794017522444564006442664306525766597161469787
17332526254247928986503718943545941228409997415431018962702806278524534
66583572866286863688798455159498253836109219126642962373937070041657442
08695618234138808452257672772438474671177397384842843716438786042706893
75975912478122192280406495103069267569084049922680543292121637319172591
79566018401847209738098004334904688848677255967871931296704384455471587
75669733723407193952257210589631928286911089116030508338233863473536922
75555776466241294432324170157283095834048906391686293082742737202863073
2131104857473817762574727003026725415471938291973
public exponent:
65537

  Validity: [From: Wed Feb 05 14:36:08 IST 2014,
               To: Mon Oct 31 14:36:08 IST 2016]
  Issuer: CN=sklm, OU=sales, O=myCompanyName, C=US
  SerialNumber: [187046468526998]

Certificate Extensions: 1
[1]: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: 37 28 bb 82 e8 02 14 68  a0 51 13 75 7a 7a 80 e2  7......h.Q.uzz..
0010: 10 0b 9b d5                                        ....
]
]

]
  Algorithm: [SHA256withRSA]
  Signature:
0000: 66 c8 b7 ca d2 35 5c 11  0d ee cd 77 57 52 47 8a  f....5.....wWRG.
0010: e0 98 bb f8 4e 43 d1 56  dd 29 24 6c 43 d7 ec 9d  ....NC.V...lC...
0020: a6 8a e9 d4 b3 b2 1b 0a  37 be 09 d7 5a 7a 26 8c  ........7...Zz..
0030: f9 6b ac d7 a4 2e a8 c3  d6 45 e6 a4 ae 6c df ad  .k.......E...l..
0040: b5 c5 db c5 b1 f5 44 2d  30 84 60 41 0b a1 77 89  ......D.0..A..w.
0050: 78 ee 59 d0 9a ea 8d 32  95 c6 26 bf 39 6b 46 67  x.Y....2....9kFg
0060: 6b 0c 65 fa 09 a1 49 0a  7b 0e fe af 20 cf d3 fd  k.e...I.........
0070: 3f 4b 55 03 4d 6f 8e ef  ca 0e e6 a0 c1 91 06 f7  .KU.Mo..........
0080: b0 6c ef 49 a4 b3 2e 4a  1f 8d 2c 0f cd f3 1e aa  .l.I...J........
0090: 28 0f 1b 51 09 fb 73 dc  79 ba 0c d6 a6 2c 65 a6  ...Q..s.y.....e.
00a0: f7 51 25 7f 7d 54 5b 19  7a 5c 3e 6c fb e9 7e 45  .Q...T..z..l...E
00b0: be 6a c8 42 22 f2 21 e7  6c c3 be 9a c2 f1 2c 64  .j.B....l......d
00c0: a5 1b bd 79 a7 c7 aa f7  5f e7 7d 76 c0 2c c4 f8  ...y.......v....
00d0: 77 48 2c 7e f7 04 a3 d4  b8 0e 99 56 2a b5 f7 b4  wH.........V....
00e0: 06 f3 b8 b7 7f d4 e3 b5  ff 35 05 fa 64 10 75 79  .........5..d.uy
00f0: 85 f1 97 bb ab ce f1 08  bc b5 d0 73 a5 34 80 5a  ...........s.4.Z

]
  }
]
```

Error response

```
Status Code : 400 Bad Request
Content-Language: en
{"code" "CTGKM6002E",
 "message": "CTGKM6002E Bad Request: Invalid user authentication
 ID or invalid request format."
 }
```

Pending Client Certificate Reject REST Service

Use **`Pending Client Certificate Reject REST Service`** to reject the certificate that is pushed to the server from a client device. This service also discards the certificate data from the IBM Security Key Lifecycle Manager database. You cannot use this certificate for secure communication.

Operation

`DELETE`

URL `https://<host>:<port>/SKLM/rest/v1/pendingClientCertificates/{uuid}`

Request

Request Parameters

Parameter	Description
host	Specify the IP address or host name of the IBM Security Key Lifecycle Manager server.
port	Specify the port number on which the IBM Security Key Lifecycle Manager server listens for requests.

Request Headers

Header name	Value
Content-Type	application/json
Accept	application/json
Authorization	SKLMAuth userAuthId=<authIdValue>
Accept-Language	Any valid locale that is supported by IBM Security Key Lifecycle Manager. For example: en or de

Path parameter

JSON object with the following specification:

Property name	Description
uuid	Specify the universal unique identifier of the certificate in the IBM Security Key Lifecycle Manager database, such as `CERTIFICATE-a3862239-a367-41ff-97a1-0ca72cfa08e8`.

Response

Response Headers

Header name	Value and description
Status Code	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.
Content-Type	`application/json`
Content-Language	Locale for the response message.

Success response body

JSON object with the following specification:

JSON property name	Description
code	Returns the code that is specified by the status property.
status	Returns the status to indicate whether the rejection of pending client certificate was successful.

Error Response Body

JSON object with the following specification:

JSON property name	Description
code	Returns the application error code.
message	Returns a message that describes the error.

Examples

Service request to reject a pending client certificate

```
DELETE https://localhost:9080/SKLM/rest/v1/pendingClientCertificates/
CERTIFICATE-4e064e39-5c15-4e29-83ab-ebd4d275e148
Content-Type: application/json
Accept : application/json
Authorization : SKLMAuth userAuthId=37ea1939-1374-4db7-84cd-14e399be2d20
Accept-Language : en
```

Success response

```
Status Code: 200 OK
{"code": "0","status": "Succeeded"}
```

Error response when there is no pending client certificate

```
Status Code : 500 Internal Server Error
{"code":"CTGKM2307E","message":"CTGKM2307E  Client certificate UUID
not found in the database: CERTIFICATE-4e064e39-5c15-4e29-83ab-
ebd4d275e148 "}
```

Replication Now REST Service

Use **Replication Now REST Service** to immediately run IBM Security Key Lifecycle Manager replication and to force a backup to be sent to the configured clones.

Operation
POST

URL https://*<host>*:*<port>*/SKLM/rest/v1/replicate/now

Request

Request Parameters

Parameter	Description
host	Specify the IP address or host name of the IBM Security Key Lifecycle Manager server.
port	Specify the port number on which the IBM Security Key Lifecycle Manager server listens for requests.

Request Headers

Header name	Value
Content-Type	application/json
Accept	application/json
Authorization	SKLMAuth userAuthId=<authIdValue>
Accept-Language	Any valid locale that is supported by IBM Security Key Lifecycle Manager. For example: en or de

Request body

JSON object with the following specification:

JSON property name	Description
replicationTargetFromConfig	Conditional. If you specify the value yes, the values for the **hostname** and **port** are taken from the configuration file. Else, you must specify the value for **hostname** and **port**.

Request body

JSON object with the following specification:

JSON property name	Description
hostname	Conditional. Specify the host name or IP of replication target. If you specify this parameter, the **port** parameter is required. The value is ignored if the value of the **replicationTargetFromConfig** parameter is yes.
port	Conditional. Specify the port number to connect to the replication clone system. If you specify this parameter, the **hostname** parameter is required. The value is ignored if the value of the **replicationTargetFromConfig** parameter is yes.

Response

Response Headers

Header name	Value and description
Status Code	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.
Content-Type	application/json
Content-Language	Locale for the response message.

Success response body

JSON array that contains JSON objects with the following specification:

JSON property name	Description
code	Returns the value that is specified by the **status** property.

Success response body

JSON array that contains JSON objects with the following specification:

JSON property name	Description
status	Returns the status message that indicates whether the replication task is run: **CTGKM2200I** Replication has been successful for the host listed. **CTGKM2201W** .Replication already in progress. **CTGKM2202E** Replication has failed for the host listed. **CTGKM2203E** Replication has failed for the host listed with a connection error. **CTGKM2204E** Replication has failed for the host listed with a validation error. **CTGKM2212E** Replication for the specified host timed out. **CTGKM2243E** Replication can only be invoked on the master machine. **CTGKM2222E** No valid replication config file exists.

Error Response Body

JSON object with the following specification:

JSON property name	Description
code	Returns the application error code.
message	Returns a message that describes the error.

Examples

Service request to run the replication task

```
POST https://localhost:9080/SKLM/rest/v1/replicate/now
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
{"hostname":"remotehost","port":"2222"}
```

Success response

```
Status Code : 200 OK
[
 { "code":"CTGKM2200I","status":" CTGKM2200I Replication successful
  for remotehost:2222" }
]
```

Service request to run the replication task without specifying the port number

```
POST https://localhost:9080/SKLM/rest/v1/replicate/now
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
{"hostname":"remotehost"}
```

Error response

```
Status Code : 400 Bad Request
{"code":"CTGKM0631E","message":"CTGKM0631E Missing required
parameter
\"  port  \" ."}
```

Replication Start REST Service

Use **Replication Start REST Service** to replicate the current IBM Security Key Lifecycle Manager active files and data across systems.

Operation
POST

URL https://*<host>*:*<port>*/SKLM/rest/v1/replicate/start

Request

Request Parameters

Parameter	Description
host	Specify the IP address or host name of the IBM Security Key Lifecycle Manager server.
port	Specify the port number on which the IBM Security Key Lifecycle Manager server listens for requests.

Request Headers

Header name	Value
Content-Type	application/json
Accept	application/json
Authorization	SKLMAuth userAuthId=<authIdValue>
Accept-Language	Any valid locale that is supported by IBM Security Key Lifecycle Manager. For example: en or de

Response

Response Headers

Header name	Value and description
`Status Code`	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.
`Content-Type`	`application/json`
`Content-Language`	Locale for the response message.

Success response body

JSON object with the following specification:

JSON property name	Description
`code`	Returns the value that is specified by the **`status`** property.

`status`	Returns the status message that indicates the success or failure of the replication task:
	`CTGKM2207W`
	`IBM Security Key Lifecycle Manager replication task is already up.`
	`CTGKM2205I`
	`IBM Security Key Lifecycle Manager replication task started successfully.`
	`CTGKM2206E`
	`IBM Security Key Lifecycle Manager replication task failed to start.`

Error Response Body

JSON object with the following specification:

JSON property name	Description
`code`	Returns the application error code.
`message`	Returns a message that describes the error.

Examples

Service request to start the replication task

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

Success response

```
Status Code : 200 OK
{"code": "CTGKM2205I","status": "CTGKM2205I IBM Security Key
Lifecycle Manager replication task started successfully."}}
```

Service request to start the replication task without specifying the configuration file

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

Error response

```
Status Code : 500 Internal Server Error
{"code": "CTGKM2222E","message": "CTGKM2222E No valid replication
config file exists."}
```

Replication Status REST Service

Use **Replication Status REST Service** to obtain information about the IBM Security Key Lifecycle Manager replication task, such as operational status and replication schedules.

Operation
GET

URL https://*<host>*:*<port>*/SKLM/rest/v1/replicate/status

Request

Request Parameters

Parameter	Description
host	Specify the IP address or host name of the IBM Security Key Lifecycle Manager server.
port	Specify the port number on which the IBM Security Key Lifecycle Manager server listens for requests.

Request Headers

Header name	Value
Content-Type	application/json
Accept	application/json
Authorization	SKLMAuth userAuthId=<authIdValue>
Accept-Language	Any valid locale that is supported by IBM Security Key Lifecycle Manager. For example: en or de

Response

Response Headers

Header name	Value and description
`Status Code`	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.
`Content-Type`	`application/json`
`Content-Language`	Locale for the response message.

Success response body

JSON array that contains JSON objects with the following specification:

JSON property name	Description
`code`	Returns the value that is specified by the `status` property.

`status`	Returns the status messages that indicate the replication status:
	CTGKM2216I
	`IBM Security Key Lifecycle Manager replication task is down.`
	CTGKM2215I
	`IBM Security Key Lifecycle Manager replication task is up.`
	CTGKM2220I
	`No successful instances of IBM Security Key Lifecycle Manager replication are taken place in this installation.`
	CTGKM2218I
	`Date and time of the last completed IBM Security Key Lifecycle Manager replication.`
	CTGKM2221I
	`No instances of IBM Security Key Lifecycle Manager replication are currently scheduled to take place.`
	CTGKM2238I
	`Scheduled time has reached. Waiting to schedule next event.`
	CTGKM2217I
	`Date and time of the next scheduled IBM Security Key Lifecycle Manager replication.`
	CTGKM2222E
	`No valid replication config file exists.`

Error Response Body

JSON object with the following specification:

JSON property name	Description
code	Returns the application error code.
message	Returns a message that describes the error.

Examples

Service request to obtain the replication status

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

Success response

```
Status Code : 200 OK
[
  {"code":"CTGKM2215I", "status":"CTGKM2215I
The IBM Security Key Lifecycle Manager replication task is up.
   Role set to: MASTER"} ,
  {"code":"CTGKM2220I", "status":"CTGKM2220I No previous successful
   replications."} ,
  { "code":"CTGKM2221I", "status":"CTGKM2221I No replication currently
    scheduled."}
  {"code":"CTGKM2222E","status": "CTGKM2222E No valid replication
   config file exists."}

]
```

Replication Stop REST Service

Use **Replication Stop REST Service** to stop the replication of current IBM Security Key Lifecycle Manager active files and data across systems.

Operation
`POST`

URL `https://<host>:<port>/SKLM/rest/v1/replicate/stop`

Request

Request Parameters

Parameter	Description
host	Specify the IP address or host name of the IBM Security Key Lifecycle Manager server.
port	Specify the port number on which the IBM Security Key Lifecycle Manager server listens for requests.

Request Headers

Header name	Value
Content-Type	application/json
Accept	application/json
Authorization	SKLMAuth userAuthId=<authIdValue>
Accept-Language	Any valid locale that is supported by IBM Security Key Lifecycle Manager. For example: en or de

Response

Response Headers

Header name	Value and description
Status Code	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.
Content-Type	`application/json`
Content-Language	Locale for the response message.

Success response body

JSON array that contains JSON objects with the following specification:

JSON property name	Description
code	Returns the value that is specified by the **status** property.

status	Returns the status message that indicates the success or failure of the replication stop task:
	CTGKM2210W
	`IBM Security Key Lifecycle Manager replication task is already down.`
	CTGKM2208I
	`.IBM Security Key Lifecycle Manager replication task stopped successfully.`
	CTGKM2209E
	`IBM Security Key Lifecycle Manager replication task failed to stop.`
	CTGKM2222E
	`No valid replication config file exists.`

Error Response Body

JSON object with the following specification:

JSON property name	Description
code	Returns the application error code.
message	Returns a message that describes the error.

Examples

Service request to stop the replication task

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

Success response

```
[
  Status Code : 200 OK
  {"code": "CTGKM2208I","status": "CTGKM2208I IBM Security Key
   Lifecycle Manager replication task has stopped successfully."}
  {"code": "CTGKM2222E","status": "CTGKM2222E No valid replication
   config file exists."}
 ]
```

Secret Key Create REST Service

Use **Secret Key Create REST Service** to create one or more symmetric keys to encrypt and decrypt data.

Operation
`POST`

URL `https://<host>:<port>/SKLM/rest/v1/keys`

Request

Request Parameters

Parameter	Description
host	Specify the IP address or host name of the IBM Security Key Lifecycle Manager server.
port	Specify the port number on which the IBM Security Key Lifecycle Manager server listens for requests.

Request Headers

Header name	Value
Content-Type	application/json
Accept	application/json
Authorization	SKLMAuth userAuthId=<authIdValue>
Accept-Language	Any valid locale that is supported by IBM Security Key Lifecycle Manager. For example: en or de

Request body

JSON object with the following specification:

JSON property name	Description
alias or **aliasRange**	Specify the key **alias** if **aliasRange** is not specified. You must also specify the value for **numOfKeys**. Specify the value for **aliasRange** if alias is not specified.
numOfKeys	Specify the number of keys to create. If you specify a value for **alias**, also specify a value for this parameter. Default value is 1.

Request body

JSON object with the following specification:

JSON property name	Description
usage	Specify the target application usage with any of the following values: **LTO** Specifies the LTO device group. **DS5000** Specifies the DS5000 device group. **GENERIC** Specifies a device family that uses the Key Management Interoperability Protocol to interact with IBM Security Key Lifecycle Manager. The GENERIC device group enables management of KMIP objects.
keyGroupUuid	Specify the UUID of the key group.

Response

Response Headers

Header name	Value and description
Status Code	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.
Content-Type	application/json
Content-Language	Locale for the response message.

Success response body

JSON array that contains JSON objects with the following specification:

JSON property name	Description
CreatedKeys	JSON array that contains JSON objects with a list of created keys. If keys are not created, an empty list is returned.
ExistingKeys	JSON array that contains JSON objects with a list of duplicate keys. If there are no duplicate keys, an empty list is returned.
FailedToCreateKeys	JSON array that contains JSON objects with a list of failed keys. If there are no failed keys, an empty list is returned.
status	This field is in the response only if a problem is encountered when you create a key.

Error Response Body

JSON object with the following specification:

JSON property name	Description
code	Returns the application error code.
message	Returns a message that describes the error.

Examples

Service request to create asymmetric key

```
POST https://localhost:9080/SKLM/rest/v1/keys
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
{"alias":"abc","numOfKeys":"2","keyGroupUuid":"KEYGROUP-720aac61-9516-4bb9-
9fa388f5db2823db","usage":"LTO"}
```

Success response

```
Status Code : 200 OK
{"Created-Keys":[{"KeyAlias":"abc0000000000000000079"},{"KeyAlias":
"abc000000000000000007a"}],"ExistingKeys":[],"FailedToCreateKeys":[]}
```

Error response

```
Status Code : 500 Internal Server Error
{"code":"CTGKM0741E","message":"Only one of alias or aliasRange parameters
can be specified"}
```

Served Data List REST Service

Use **Served Data List REST Service** to query the database and list the served key data. For example, you might list which devices were served a specific key, or list the keys that were served to a specific device.

Served Data List REST Service supports pagination. The request parameters, such as offset and count, are used for pagination. For example, to retrieve the first 10 records for the list, set **offset = 1** and **count = 10**. To retrieve the next 10 records, set **offset = 2** and **count = 10**. If you do not specify values for pagination parameters, the first 2000 records are returned.

Operation

GET

URL

Retrieve all the served key data:

https://<*host*>:<*port*>/SKLM/rest/v1/servedData

Note: Returns 2000 records.

Retrieve all the served key data when you specify a few parameters:

```
https://<host>:<port>/SKLM/rest/v1/
servedData?kmipClientCertUUID=<clientCertUUID>&dateBefore=<date>
&dateAfter=<date>
```

Note: Returns 2000 records.

Retrieve all the served key data when you specify all the parameters:

```
https://<host>:<port>/SKLM/rest/v1/
servedData?volser=<VolumeSerialNumber>&attributeName=<attrName>
&attributeValue=<attrvalue>&dateBefore=<date>&dateAfter=<date>&usage
=<devicetype>&serialNumber=<deviceSerialNumber>&kmipClientCertUUID
=<clientCertUUID>
```

Note: Returns 2000 records.

To retrieve a specific list with pagination:

```
https://<host>:<port>/SKLM/rest/v1/
servedData?volser=<VolumeSerialNumber>&attributeName=<attrName>
&attributeValue=<attrvalue>&dateBefore=<date>&dateAfter=<date>&usage=
<devicetype>&serialNumber=<deviceSerialNumber>&kmipClientCertUUID=
<clientCertUUID>&offset=<offset>&count=<count>
```

Request

Request Parameters

Parameter	Description
host	Specify the IP address or host name of the IBM Security Key Lifecycle Manager server.
port	Specify the port number on which the IBM Security Key Lifecycle Manager server listens for requests.

Request Headers

Header name	Value
Content-Type	`application/json`
Accept	`application/json`
Authorization	`SKLMAuth userAuthId=<authIdValue>`
Accept-Language	Any valid locale that is supported by IBM Security Key Lifecycle Manager. For example: `en` or `de`

Query parameters

Parameter name	Description
attributeName	Optional. **alias1** Specify a default alias for a certificate that is used by a 3592 tape drive or a DS8000 Turbo drive. Not used for an LTO tape drive or DS5000 storage server. **3592 tape drive** The value is optional for a 3592 tape drive and specifies the primary certificate that the device in the 3592 device family uses. If this attribute is not specified, the partner default certificate is used, as specified by a table entry for the device group in the IBM Security Key Lifecycle Manager database. **DS8000 Turbo drive** The value is optional for a DS8000 Turbo drive and matches the label "Primary certificate for image" in the graphical user interface panels for a DS8000 Turbo drive. Use **Device Group Attribute List REST Service** and **Device Group Attribute Update REST Service** to view and change the value. This value was previously stored in the obsolete configuration parameter **drive.default.alias1**. **alias2** Used for a 3592 tape drive or a DS8000 Turbo drive. Not used for an LTO tape drive or DS5000 storage server. **3592 tape drive** This attribute specifies a default alternative alias for a 3592 tape drive. This value can be the same, or different from the value that is specified for the primary certificate. The value specifies the secondary certificate that the device in the 3592 device family uses if the primary certificate is not available. If this attribute is not specified, the partner default certificate is used, as specified by a table entry for the device group in the IBM Security Key Lifecycle Manager database. **DS8000 Turbo drive** For a device in the DS8000 device family, the value specifies a secondary certificate that is available for use. For example, you might use this certificate to unlock a DS8000 Turbo drive in the case of a dead-lock condition. Use **Device Group Attribute List REST Service** and **Device Group Attribute Update REST Service** to view and change the value. This value was previously stored in the obsolete configuration parameter **drive.default.alias2**. **dki** Data key identifier, used only for an LTO tape drive.
attributeValue	Optional. Identifies the served data. For example, if attributeName is alias1, then attributeValue might be cert1.

Query parameters

Parameter name	Description
dateBefore	Optional. If you specify only this date, list the audits that are made before this date. Hyphens are required in the date value. To list audits that are made between the before and after dates, specify both values. Format for the date is YYYY-MM-DD.
dateAfter	Optional. If you specify only this date, list the audits that are made after this date. Hyphens are required in the date value. To list audits that are made between the before and after dates, specify both values. Format for the date is YYYY-MM-DD.
usage	Optional. Specify one of the following values: **LTO** Specifies the LTO device family. **3592** Specifies the 3592 device group. **DS5000** Specifies the DS5000 device group. **DS8000** Specifies the DS8000 device group. **BRCD_ENCRYPTOR** Specifies the BRCD_ENCRYPTOR device group that is in the LTO device family. **ONESECURE** Specifies the ONESECURE device group that is in the DS5000 device family. **userdevicegroup** Specifies a user-defined group that is based on a supported device family.
volser	Optional. Specify the volume and serial number of a tape cartridge.
kmipClientCertUUID	Optional. Specify UUID of the KMIP client certificate.
serialNumber	Optional. Specify the device serial number.
offset	Optional. Specify the page number from which the records are displayed based on the value that you specify for **count**.
count	Optional. Specify the number of records to display on the page that you specified with **offset**. The count must not exceed 2000 records.

Response

Response Headers

Header name	Value and description
Status Code	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.
Content-Type	application/json
Content-Language	Locale for the response message.

Success response body

JSON array that contains JSON objects with the following specification

JSON property name	Description
Device uuid	Returns the universal unique identifier of the device.
Serial Number	Returns the serial number of the device as an ASCII string.
Volume Serial Number	Returns the volume and serial number of the tape cartridge.
World wide name	Returns the name of a device.
Key alias 1	Returns the default key alias.
Key alias 2	Returns the alias of the key served.
TimeStamp	Returns the time stamp when the key was last served.
Data Key Identifier (dki)	Returns the data key identifier.
Attributes	Returns one or more device attributes.
Device Group Name	Returns the device type.
Kmip Client Certificate UUID	Returns the universal unique identifier of KMIP client certificate.

Error Response Body

JSON object with the following specification:

JSON property name	Description
code	Returns the application error code.
message	Returns a message that describes the error.

Examples

Service request to list key served data information

```
GET https://localhost:9080/SKLM/rest/v1/servedData?offset=1&count=2
Content-Type: application/json
Accept : application/json
Authorization : SKLMAuth userAuthId=37ea1939-1374-4db7-84cd-14e399be2d20
Accept-Language : en
```

Success response

```
Status Code : 200 OK
Content-Language: en

[
 {
  "Device uuid" :   "uuid103",
  "Serial Number":  "null",
  "Volume Serial Number": "TEST",
  "World wide name":  "null",
  "Key alias 1":  "null",
  "Key alias 2":  "null",
  "TimeStamp":    "Thursday, June 26, 2014 5:44:19 AM Eastern Daylight
   Time",
   "Data Key Identifier (dki)": "null"
  "Attributes":    "Attributes":   "null",
  "Device Group Name": "UNSET"
 },
 {
  "Device uuid" : "uuid101",
   "Serial Number":   "null",
  "Volume Serial Number":  "null",
  "World wide name":  "null",
  "Key alias 1":  "dsk00000000000000000e",
  "Key alias 2":  "null",
  "TimeStamp":    "Thursday, June 26, 2014 5:44:19 PM Eastern Daylight
   Time",
   "Data Key Identifier (dki)": "null",
  "Attributes":   "null",
  "Device Group Name": "UNSET"
 }
]
```

Error response

```
Status Code : 400 Bad Request
Content-Language: en
{"code":"CTGKM6002E","message":"CTGKM6002E Bad Request: Invalid user
authentication ID or invalid request format."}
```

Truststore Certificate List REST Service

Use **`Truststore Certificate List REST Service`** to list certificates that are present in the IBM Security Key Lifecycle Manager internal truststore.

Operation

`GET`

URL https://*<host>*:*<port>*/SKLM/rest/v1/trustStoreCertificates

https://*<host>*:*<port>*/SKLM/rest/v1/trustStoreCertificates?alias=<aliasName>

Request

Request Parameters

Parameter	Description
host	Specify the IP address or host name of the IBM Security Key Lifecycle Manager server.
port	Specify the port number on which the IBM Security Key Lifecycle Manager server listens for requests.

Request Headers

Header name	Value
Content-Type	application/json
Accept	application/json
Authorization	SKLMAuth userAuthId=<authIdValue>
Accept-Language	Any valid locale that is supported by IBM Security Key Lifecycle Manager. For example: en or de

Query parameter

JSON object with the following specification:

JSON property name	Description
alias	Optional. Specify a unique name for the certificate. To list all certificates, do not specify an **alias**.

Response

Response Headers

Header name	Value and description
Status Code	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.
Content-Type	application/json
Content-Language	Locale for the response message.

Success response body

JSON array that contains JSON objects with the following specification:

JSON property name	Description
alias	Returns the alias name of the truststore certificate.
key state	Returns the state of the certificate, such as ACTIVE.

Success response body

JSON array that contains JSON objects with the following specification:

JSON property name	Description
issuer name	Returns the name of the issuer of the certificate.
subject name	Returns the subject name of the certificate.
activation date	Returns the activation date of the certificate.
creation date	Returns the certificate creation date.
expiration date	Returns the certificate expiration date.
trusted	Indicates whether the certificate is trusted by returning the following values: **1** Trusted **0** Not trusted.
serial number	Returns the certificate serial number.
hash value	Returns the digest value for the managed object, that is, certificate.

Error Response Body

JSON object with the following specification:

JSON property name	Description
code	Returns the application error code.
message	Returns a message that describes the error.

Examples

Service request to obtain truststore certificate list

```
GET https://localhost:9080/SKLM/rest/v1/trustStoreCertificates?
alias=ibmdiskds5000
Content-Type: application/json
Accept : application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
```

Success response

```
Status Code : 200 OK
    [
       {
           "alias": "ibmdiskdriveds5000",
           "key state": "ACTIVE",
           "issuer name": "O=ibmDiskDrive,OU=StorageHardware,
      CN=DS5000StorageRoot,C=US",
           "subject name": "O=ibmDiskDrive,OU=StorageHardware,
      CN=DS5000StorageRoot,C=US",
           "activation date": "2/15/10 9:20:45 PM India Standard
            Time",
           "creation date": "6/27/13 1:06:33 AM India Standard
            Time",
           "expiration date": "2/10/30 9:20:45 PM India Standard
            Time",
           "trusted": "1",
           "serial number": "0",
           "hash value": "0000: fa 65 6f f6 37 2c 0f 0e 42 dc b9 2d
```

```
  a9 5f 8b 01 .eo.7...B....... 0010: 2d ac 0c 73 bc c8 d0
  a9 62 0d 33 e3 e0 d8 37 5c ...s....b.3...7. "
   }
]
```

Update Config Property REST Service

Use **Update Config Property REST Service** to update one or more properties in the SKLMConfig.properties file that controls the IBM Security Key Lifecycle Manager server operations.

Operation
PUT

URL https://*<host>*:*<port>*/SKLM/rest/v1/configProperties

Request

Request Parameters

Parameter	Description
host	Specify the IP address or host name of the IBM Security Key Lifecycle Manager server.
port	Specify the port number on which the IBM Security Key Lifecycle Manager server listens for requests.

Request Headers

Header name	Value
Content-Type	application/json
Accept	application/json
Authorization	SKLMAuth userAuthId=<authIdValue>
Accept-Language	Any valid locale that is supported by IBM Security Key Lifecycle Manager. For example: en or de

Request body

JSON property name	Description
<Property names>	Specify the configuration property names that you want to update. You can specify multiple comma-separated properties.

Response

Response Headers

Header name	Value and description
`Status Code`	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.
`Content-Type`	`application/json`
`Content-Language`	Locale for the response message.

Success response body

JSON object with the following specification:

JSON property name	Description
`property`	Returns the name of the property that is updated.
`status`	Returns the status to indicate the configuration file updates.

Note: The success response code `200 OK` is returned even if the property you requested is not found. An appropriate message is returned in the response body.

Error Response Body

JSON object with the following specification:

JSON property name	Description
`code`	Returns the application error code.
`message`	Returns a message that describes the error.

Examples

Service request to update a single server configuration property

```
PUT https://localhost:9080/SKLM/rest/v1/configProperties
Content-Type: application/json
Accept : application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
Accept-Language : en
{ "KMIPListener.ssl.port" : "10000"}
```

Success response

```
Status Code : 200 OK
Content-Language: en
[{"property":"KMIPListener.ssl.port","status":"CTGKM0607I Update
successful, server restart requi-red for change to take effect"}]
```

Error response

```
Status Code : 400 Bad Request
Content-Language: en
{"code" : "CTGKM6002E", "message" : "CTGKM6002E Bad Request:
Invalid user authentication ID or invalid request format"}
```

Service request to update multiple server configuration properties

```
PUT https://localhost:9080/SKLM/rest/v1/configProperties
Content-Type: application/json
Accept : application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
Accept-Language : en
 {"fips" : "on", "KMIPListener.ssl.port" : "5678"}
```

Success response

```
Status Code : 200 OK
Content-Language: en
[{"property":"KMIPListener.ssl.port","status":"CTGKM0607I Update
successful, server restart requi-red for change to take effect"},{"
property":"fips","status":"CTGKM0606I Update successful, change will
take effect immediately"}]
```

Error response

```
Status Code : 400 Bad Request
Content-Language: en
{"code" : "CTGKM6002E", "message" : "CTGKM6002E Bad Request:
Invalid user authentication ID or invalid request format"}
```

Update Replication Config Property REST Service

Use **Update Replication Config Property REST Service** to update one or more properties in the ReplicationSKLMConfig.properties configuration file to control the replication operation.

Operation

PUT

URL https://*<host>*:*<port>*/SKLM/rest/v1/replicationConfigProperties

Request

Request Parameters

Parameter	Description
host	Specify the IP address or host name of the IBM Security Key Lifecycle Manager server.
port	Specify the port number on which the IBM Security Key Lifecycle Manager server listens for requests.

Request Headers

Header name	Value
Content-Type	application/json
Accept	application/json
Authorization	SKLMAuth userAuthId=<authIdValue>
Accept-Language	Any valid locale that is supported by IBM Security Key Lifecycle Manager. For example: en or de

Request body

JSON property name	Description
<propertyNames>	Specify the replication configuration property names and values that you want to update. You can specify multiple comma-separated properties.

Response

Response Headers

Header name	Value and description
Status Code	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.
Content-Type	application/json
Content-Language	Locale for the response message.

Success response body

JSON object with the following specification:

JSON property name	Description
property	Returns the name of the property that is updated.
status	Returns the update status to indicate whether the configuration property is updated with an appropriate message.

Note: The success response code 200 OK is returned even if the property you requested is not found. An appropriate message is returned in the response body.

Error Response Body

JSON object with the following specification:

JSON property name	Description
code	Returns the application error code.
message	Returns a message that describes the error.

Examples

Service request to update a single configuration property

```
PUT https://localhost:9080/SKLM/rest/v1/replicationConfigProperties
Content-Type: application/json
Accept : application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
Accept-Language : en
{ "backup.ClientIP1":"9.118.40.184"}
```

Success response

```
Status Code : 200 OK
Content-Language: en
[{"property":"backup.Client","status":"CTGKM0607I Update
successful, server restart required for change to take effect"}]
```

Error response

```
Status Code : 400 Bad Request
Content-Language: en
{"code" : "CTGKM6002E", "message" : "CTGKM6002E Bad Request:
Invalid user authentication ID or invalid request format"}
```

Service request to update multiple configuration properties

```
PUT https://localhost:9080/SKLM/rest/v1/replicationConfigProperties
Content-Type: application/json
Accept : application/json
Authorization: SKLMAuth userAuthId=139aeh34567m
Accept-Language : en
 {"backup.ClientIP1" : "9.118.40.184", "replication.role" : "master"}
```

Success response

```
Status Code : 200 OK
Content-Language: en
[{"property":"backup.ClientIP1","status":"CTGKM0607I Update
successful, server restart required for change to take effect"},
{"property":"replication.role","status":"CTGKM0606I Update
successful, change will take effect immediately"}]
```

Error response

```
Status Code : 400 Bad Request
Content-Language: en
{"code" : "CTGKM6002E", "message" : "CTGKM6002E Bad Request:
Invalid user authentication ID or invalid request format"}
```

Command-line interface

The IBM Security Key Lifecycle Manager command-line interface provides commands to match the function in the IBM Security Key Lifecycle Manager graphical user interface. The commands use the wsadmin interface that the WebSphere® Application Server provides.

Note:

- The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager. Use the REST interfaces instead.
- All references to the **alias** property of cryptographic keys and certificates in the graphical user interface, command-line interface, and REST interface will be deprecated in the later versions of IBM Security Key Lifecycle Manager.

wsadmin commands

You can run **wsadmin** commands in batch or interactive mode, in a script, or from a command prompt by using the **wsadmin -c** command.

Note: To successfully run commands on the IBM Security Key Lifecycle Manager command-line interface, which uses **wsadmin** commands, you must initially log in as root on systems such as Linux or AIX.

Use one of these formats for a command:

commandName
: Starts a command that does not require an argument.

commandName *options*
: Starts a command with the specified options. Use this syntax to enter interactive mode if `-interactive` mode is included in the options.

  Note:

  - Empty values "" for attributes are allowed only in interactive mode.
  - In interactive mode, running a command with a space at the end of a value returns an error message. Be careful to not type extra spaces in commands when you use interactive mode.

The **wsadmin** tool supports two scripting languages: JACL and Jython. Because the Jython syntax for **wsadmin** is the strategic direction for WebSphere Application Server administrative automation, these examples use the Jython syntax.

The Jython syntax has a parameter and a value:

```
AdminTask.tklmConfigList()
AdminTask.tklmKeyStoreList('[-v y]')
```

To enable Jython, specify the **`wsadmin -lang jython`** command-line parameter. For example, change directory to the `bin` directory in *`WAS_HOME`*:

- Windows systems: *`drive`*`:\Program Files (x86)\IBM\WebSphere\AppServer\bin`
- Other systems such as Linux: `/opt/IBM/WebSphere/AppServer`

Then, type:

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

Format output by using `print` for readability

In Jython, use `print` before the **`AdminTask`** command to format the output for readability. If you do not use `print`, output characters have a corrupted appearance in some languages.

Do not add `print` to an **`AdminTask`** command in a script file.

Session security by using wsadmin

There is no automatic timeout for an idle **wsadmin** session. To maintain a secure session, you must manually close the session. Ensure that your system has a password-protected screen lock.

IBM Security Key Lifecycle Manager commands by using wsadmin

To get a list of all command groups by using **wsadmin**, type:

```
print AdminTask.help ("-commandGroups")
```

IBM Security Key Lifecycle Manager supports these command groups:

TKLMBackupCommands
Back up and restore IBM Security Key Lifecycle Manager data.

TKLMConfigurationCommands
Configure the IBM Security Key Lifecycle Manager application.

TKLMDefaultRolloverCommands
Manage certificates and key groups that are specified in a default rollover list.

TKLMDeviceCommands
Manage devices such as tape drives.

TKLMDeviceGroupCommands
Manage device groups such as the LTO device group.

TKLMGroupCommands
Manage groups of keys.

TKLMInfoCommands
Obtain more information such as the IBM Security Key Lifecycle Manager version number.

TKLMKeyCertManagementCommands
Manage keys and certificates.

TKLMKeyServerCommands
Manage an internal component that is called the keyserver.

TKLMKeyStoreCommands
Manage the IBM Security Key Lifecycle Manager keystore.

TKLMKMIPCommands
Manage KMIP objects.

TKLMMachineAffinityCommands
Manage the association of disk drives to DS5000 storage servers.

TKLMPendingAutoCommands
Manage the acceptance or rejection of devices in a list of pending devices to which IBM Security Key Lifecycle Manager might serve keys or certificates upon request.

To see all the commands in a group such as the certificate management commands group, type:

```
wsadmin>print AdminTask.help ("TKLMKeyCertManagementCommands")
```

To see the parameters for an individual command within a group, such as the **tklmCertCreate** command, type:

```
wsadmin>print AdminTask.help ("tklmCertCreate")
```

To extract and print the help for the command, type:

```
wsadmin>s = AdminTask.help("tklmCertCreate")
wsadmin>print s
```

Larger timeout interval for command processing

To ensure adequate time to run a command that requires significant processing, you might set a larger default timeout interval. To change the value, set the **com.ibm.SOAP.requestTimeout** parameter in the *WAS_HOME*/profiles/KLMProfile/properties/soap.client.props property to a larger value, or set the value to zero for no timeout.

Syntax examples

To specify attributes within a list of parameters, use double quotation marks (") and braces {} as delimiters. For example:

```
print AdminTask.tklmDeviceUpdate
 ('[-uuid DEVICE-64c588ad-5ed8-4934-8c84-64cb9e11d990
  -attributes "{aliasTwo myPartner99}"]')
```

To avoid syntax errors in commands that you type at a prompt, do not copy and paste auto-generated commands into your command statements.

This example creates a group that has a name that contains spaces:

```
print AdminTask.tklmGroupCreate
 ('[-name "my Key Group" -type keygroup -usage LTO]')
```

To specify a value with more than one word for an attribute, use additional braces as delimiters. For example:

```
print AdminTask.tklmCertUpdate
 ('[-uuid CERTIFICATE-33fc26e-5fb5a0e66143
   -usage DS8000 -attributes "{information {new information}}"]')
```

Backslash character on Windows systems

wsadmin is a Java process that uses the backslash (\) as an escape character. On Windows systems, modify your specification of the path to a file by changing the backslash to a forward slash (/), or by using two backslash characters.

For example:

```
wsadmin>print AdminTask.tklmBackupRun
 ('[-backupDirectory c:\\tklm -description testBackUp
 -databaseBackupDirectory c:\\tklmtempdir -password password]')
```

Backslash in multi-line command entry

To avoid overwriting the right margin of printed output, these command-line examples are arbitrarily divided into lengths that fit on a printed page. They do not show an escape or continuation character.

In actual commands, however, you might use the backslash (\) as an escape or continuation character. For example:

```
print AdminTask.tklmBackupList \
 ('[-backupDirectory C:\\tipbak1\\tklmbackup1 -v y]')
```

The escape character cannot occur inside delimited values. A statement is not valid that places the backslash between a parameter such as -backupDirectory and its value. For example, this statement is not valid:

```
print AdminTask.tklmBackupList \
 ('[-backupDirectory \
  C:\\tipbak1\\tklmbackup1 -v y]')
```

Parameter error messages

If you incorrectly type the name of a parameter in a command that you run, an error message returns the incorrect string.

For example, suppose that you type -uui by mistake, instead of typing -uuid as a parameter. The return message is like the following:

```
wsadmin>AdminTask.tklmDeviceDelete('[-uui adsflk]')
  WASX7015E: Exception running command:
 "AdminTask.tklmDeviceDelete('[-uui adsflk]')"; exception information:
  com.ibm.ws.scripting.ScriptingException: WASX8009E: Invalid parameter: uui
```

Examine the parameters that you typed, by using the invalid parameter information as a guide. Make the appropriate correction. Then, try the command again.

Permissions required for commands

You must have the appropriate permissions to successfully issue a command.

A command requires permission to one or more of these actions:

Table 1. Permissions for actions

Permission	**Enables these actions**	**Unrelated to device groups**	**Associated with device groups**
klmCreate	Create but not view, modify, or delete objects		✓
klmDelete	Delete objects, but not modify or create objects		✓
klmGet	Export a key or certificate for a client device.		✓
klmModify	Modify objects, but not create or delete objects.		✓
klmView	View objects, but not create, delete, or modify objects. For example, you must have this permission to see the tasks that you want to do on the graphical user interface.		✓
klmAdminDeviceGroup	Administer (create, view, delete) a new device group	✓	
klmAudit	View audit data by using the **tklmServedDataList** command	✓	
klmBackup	Create and delete a backup of IBM Security Key Lifecycle Manager data	✓	
klmConfigure	Read or change IBM Security Key Lifecycle Manager configuration properties, or act on SSL certificates	✓	
klmRestore	Restore a previous backup copy of IBM Security Key Lifecycle Manager data	✓	

A command might also require permission to at least one of these device groups:

Table 2. Device groups

Permission	Allows actions on these objects
`LTO`	LTO device family
`TS3592`	3592 device family
`DS5000`	DS5000 device family
`DS8000`	DS8000 device family
`BRCD_ENCRYPTOR`	BRCD_ENCRYPTOR device group
`ONESECURE`	ONESECURE device group
`ETERNUS_DX`	`ETERNUS_DX` device group
`XIV`	`XIV` device group
`GENERIC`	Objects that use the Key Management Interoperability Protocol.
`userdevicegroup`	A new user-defined instance such as `myLTO` that is based on a predefined device family such as LTO.

tklmBackupGetProgress

Use the **`tklmBackupGetProgress`** command to determine the current phase of a backup task that is running.

Note: The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager. Use the REST interfaces instead.

Purpose

Use this command to determine the current phase of a backup task that is running.

The command returns an integer value:

0	IDLE
1	INITIALIZE
2	BACKUP_CONFIG_FILES
3	BACKUP_KEYSTORES
4	BACKUP_DATABASE
5	CREATE_BACKUP_JAR
6	CLEANUP
7	COMPLETED

Permissions

Your role must have a permission to back up files.

Syntax

tklmBackupGetProgress

Parameters

There are no parameters.

Example

This Jython-formatted command returns an integer value that indicates the current phase in the backup task.

```
print AdminTask.tklmBackupGetProgress()
```

tklmBackupGetRestoreProgress

Use the **`tklmBackupGetRestoreProgress`** command to determine the current phase of a restore task that is running.

Note: The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager. Use the REST interfaces instead.

Purpose

Use this command to determine the current phase of a restore task that is running. Some phases are brief. You might not be able to observe their return.

The command returns an integer value:

0	IDLE
1	INITIALIZE
2	CREATE_TEMP_DIR
3	RESTORE_CONFIG_FILES
4	RESTORE_KEYSTORES
5	RESTORE_DATABASES
6	CLEANUP
7	COMPLETED

Permissions

Your role must have a permission to restore files.

Syntax

tklmBackupGetRestoreProgress

Parameters

There are no parameters.

Example

This Jython-formatted command returns an integer value that indicates the current phase in the restore task.

```
print AdminTask.tklmBackupGetRestoreProgress()
```

tklmBackupGetRestoreResult

Use the **tklmBackupGetRestoreResult** command to determine the success or failure of a completed restore task.

Note: The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager. Use the REST interfaces instead.

Purpose

Use this command to determine the success or failure of a completed restore task.

The command returns these values:

-1 Status of the task cannot be determined. The task is not run since the last time the IBM Security Key Lifecycle Manager server started.

0 The task succeeded.

1 The task failed.

Permissions

Your role must have a permission to restore files.

Syntax

tklmBackupGetRestoreResult

Parameters

There are no parameters.

Example

This Jython-formatted command returns an integer value that describes the success or failure of a completed restore task.

```
print AdminTask.tklmBackupGetRestoreResult()
```

tklmBackupGetResult

Use the **tklmBackupGetResult** command to determine the success or failure of the most recent backup task.

Note: The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager. Use the REST interfaces instead.

Purpose

Use this command to determine the success or failure of the most recent backup task.

The command returns these values:

-1 Status of the task cannot be determined. The task is not run since the last time the IBM Security Key Lifecycle Manager server started.

0 The task succeeded.

1 The task failed.

2 Backup operation succeeded with a warning.

Permissions

Your role must have a permission to back up files.

Syntax

tklmBackupGetResult

Parameters

There are no parameters.

Example

This Jython-formatted command returns an integer value that describes the success or failure of the most recent backup task.

```
print AdminTask.tklmBackupGetResult()
```

tklmBackupIsRestoreRunning

Use the **`tklmBackupIsRestoreRunning`** command to determine whether the restore task is running.

Note: The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager. Use the REST interfaces instead.

Purpose

Use this command to determine whether the restore task is running. Only one backup or restore task can run at any time.

The command returns these values:

true The task is running.

false The task is not running.

Permissions

Your role must have permissions to back up or to restore files.

Syntax

tklmBackupIsRestoreRunning

Parameters

There are no parameters.

Example

This Jython-formatted command determines whether the restore task is running.

```
print AdminTask.tklmBackupIsRestoreRunning()
```

tklmBackupIsNeeded

Use the **`tklmBackupIsNeeded`** command to determine the keys or certificates that are not yet backed up.

Note: The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager. Use the REST interfaces instead.

Purpose

Use this command to determine the keys or certificates that are not yet backed up. The result is one of these messages:

- All keys and certificates are backed up.
- There are keys and certificates which have not been backed up. Make a backup.

Permissions

No specific permissions are needed.

Syntax

tklmBackupIsNeeded

Parameters

There are no parameters.

Example

This Jython-formatted command returns a message that describes whether any keys and certificates require backup.

```
print AdminTask.tklmBackupIsNeeded()
```

tklmBackupIsRunning

Use the **`tklmBackupIsRunning`** command to determine whether the backup task is running.

Note: The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager. Use the REST interfaces instead.

Purpose

Use this command to determine whether the backup task is running. Only one backup or restore task can run at a time.

The command returns these values:

true The task is running.

false The task is not running.

Permissions

Your role must have permissions to back up or to restore files.

Syntax

tklmBackupIsRunning

Parameters

There are no parameters.

Example

This Jython-formatted command determines whether the backup task is running.

```
print AdminTask.sklmBackupIsRunning()
```

tklmBackupList

Use the **`tklmBackupList`** command to list the backup files in the specified directory.

Note: The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager. Use the REST interfaces instead.

Purpose

Use this command to list the backup files in the specified directory.

Permissions

Your role must have permissions to back up or to restore files.

Syntax

tklmBackupList -backupDirectory *directoryname* **-v** {y| n}

Parameters

`-backupDirectory`
A directory that has JAR files with backup data for IBM Security Key Lifecycle Manager.

If you specify no path, the command appends the path that is specified by the tklm.backup.dir property in the `SKLMConfig.properties` file.

`-v [y | n]`
Verbose. The default is `n`, or no extra information. To list more information, specify `y` (for yes):

```
-v y
```

Example

This Jython-formatted command lists the jar files in a specified backup directory, and more information about the files.

```
print AdminTask.tklmBackupList
 ('[-backupDirectory C:\\tipbak1\\tklmbackup1 -v y]')
```

tklmBackupRun

Use the **`tklmBackupRun`** command to run the backup task.

Note: The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager. Use the REST interfaces instead.

Purpose

Use this command to run the backup task. Only one backup or restore task can run at a time. Ensure that there is sufficient disk space available to contain the backup files.

The time stamp on a backup file has a Greenwich Mean Time (GMT) offset represented in RFC 822 format. The file name contains a +*hhmm* or -*hhmm* element to specify a timezone ahead of or behind GMT. For example, a file name might be `sklm_v2.5.0.0_20100123144220`**`-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.

The command returns these values:

-1
State is unknown. The task is not run since the last time the IBM Security Key Lifecycle Manager server started.

0
The task succeeded.

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.

1
The task failed.

2
Backup operation succeeded with a warning.

Permissions

Your role must have a permission to back up files.

Syntax

tklmBackupRun -backupDirectory *tklmbackupdir* **-description** *backupfileinformation* **-databaseBackupDirectory** *databasebackupdir* **-password** *dataencryptionpassword***-replica**{y| n}

Parameters

`-backupDirectory`
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_HOME*/backup directory.
- If you specify a relative path (not suggested) such as mybackupdir, the backup is created in the *WAS_HOME*/profiles/KLMProfile/*mybackupdir* directory.
- IBM Security Key Lifecycle Manager can create a backup file in any directory for which the operating system superuser has permission to write the file. The superuser is Administrator on Windows systems or root on systems such as Linux or AIX.
- Do not create the backup file in the same directory that contains the database backup.

-databaseBackupDirectory

Optional. A directory in the IBM Security Key Lifecycle Manager database that contains temporary backup data for IBM Security Key Lifecycle Manager. If no parameter is specified, the directory that is used is the value of the **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.

You cannot set the value of this property by using the graphical user interface. For example, you might change the default value if you determine that another location is required to provide extra disk space for temporary database backup.

-description

More information about the purpose or use of the backup file.

-password

Required. A password that is used to encrypt the data in the backup file. The value can range between a minimum of 6 and a maximum of 32 characters.

You can use a different password for each backup file. When you restore a file, you must be able to provide the password that was used to encrypt the data in that file during the backup task.

-replica

Optional. The default is n. Specifies whether this backup is taken for replication. If you specify y, the replication configuration file is not backed up.

Example

This Jython-formatted command runs the backup task by using the directory that is specified in the **tklm.backup.dir** property in the SKLMConfig.properties file.

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

tklmBackupRunRestore

Use the **tklmBackupRunRestore** command to restore from an existing backup file.

Note: The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager. Use the REST interfaces instead.

Purpose

Use this command to restore from an existing backup file. Only one backup or restore task can run at any given time. Before you begin, obtain the password that was used to create the backup the file that you intend to use. Before you start a restore task, isolate the system for maintenance. You must restart the IBM Security Key Lifecycle Manager server immediately after the restore occurs. Verify the environment before you bring the IBM Security Key Lifecycle Manager server back into production.

The command returns these values:

-1
State is unknown. The task has not run since the last time the IBM Security Key Lifecycle Manager server started.

0
The task succeeded.

1
The task failed.

Permissions

Your role must have a permission to restore files.

Syntax

tklmBackupRunRestore -backupFilePath *pathandfilename* **-password** *dataencryptionpassword*

Parameters

The parameters include:

-backupFilePath
Required. The full path and file name of a file containing backup data for IBM Security Key Lifecycle Manager. To determine this directory, you might examine the value of the tklm.backup.dir property in the `SKLMConfig.properties` file.

-password
Required. You can use a different password for each backup file. When you restore from a given backup file, you must be able to provide the password used to encrypt the data in that file during the backup task.

Example

This Jython-formatted command restores password-encrypted backup data for IBM Security Key Lifecycle Manager.

```
print AdminTask.tklmBackupRunRestore
 ('[-backupFilePath /opt/mysklmbackups/sklm_v2.5.0.0_20130705235417-1200_backup
  -password myBackupPwd]')
```

tklmCertAttributeUpdate

Use the **tklmCertAttributeUpdate** command to update certificate metadata that are Key Management Interoperability Protocol attributes in the database.

Note: The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager. Use the REST interfaces instead.

Purpose

Use this command to update certificate metadata that are Key Management Interoperability Protocol attributes in the database.

Permissions

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

Syntax

tklmCertAttributeUpdate -uuid *certuuid* **-operation** {add | replace | delete} **-index** *indexofvalue* **-attrName** *attributename* **-attrValue** {*keyvaluepair* } {*keyvaluepair* }

Parameters

-attrName

Required. Specify the name that is used to identify or locate the attribute pair as an object.

Note: Do not use an asterisk (*) or question mark (?) as a character in a Key Management Interoperability Protocol attribute. These wildcard characters are reserved for future use.

applicationSpecificInformation

Specifies application namespace information as a Key Management Interoperability Protocol attribute.

contactInformation

Specifies contact information as a Key Management Interoperability Protocol attribute.

cryptoParams *cryptoparameter1***,** *cryptoparameterN*

Specifies the cryptographic parameters that are used for cryptographic operations by using the object. This attribute is a Key Management Interoperability Protocol attribute.

customAttribute

Specifies a custom attribute in string format as a Key Management Interoperability Protocol attribute. Client-specific attributes must start with the characters "x-" (x hyphen) and server-specific attributes must start with "y-" (y hyphen).

link

Specifies the link from one managed cryptographic object to another, closely related target managed cryptographic object. This attribute is a Key Management Interoperability Protocol attribute.

name

Specifies the name that is used to identify or locate the object. This attribute is a Key Management Interoperability Protocol attribute.

objectGroup

Specifies one or more object group names of which this object might be part. This attribute is a Key Management Interoperability Protocol attribute.

-attrValue

Specify one or more of these key value pairs:

applicationSpecificInformation *applicationIDstring*
Specifies application namespace information as the value of *applicationIDstring*.

NAMESPACE
Application namespace.

INFO
Application namespace information.

contactInformation *contactstring*
Specifies contact information as the value of *contactstring*. This attribute is a Key Management Interoperability Protocol attribute.

VALUE
Contact information.

cryptoParams *cryptoparameter1, cryptoparameterN*
Specifies the cryptographic parameters that are used for cryptographic operations by using the object. This attribute is a Key Management Interoperability Protocol attribute.

MODE
CBC, ECB, PCBC, CFB, OFB, CTR, CMAC, CCM, GCM, CBC_MAC, XTS, AES_KEY_WRAP_PADDING, NIST_KEY_WRAP, X9_102_AESKW, X9_102_TDKW, X9_102_AKW1, X 9_102_AKW2

PAD
NONE, OAEP, PKCS5, SSL3, ZEROS, ANSI_X9_23, ISO_10126, PKCS1_ V1_5, X9_31, PSS

HASH
MD2, MD4, MD5, SHA1, SHA224, SHA256, SHA384, SHA512

ROLE
BDK, CVK, DEK, MKAC, MKSMC, MKSMI, MKDAC, MKDN, MKCP, MKOTH, KEK, MAC1660 9, MAC97971, MAC97972, MAC97973, MAC97974, MAC97975, ZPK, PVKIBM, PVKPVV, PVKOTH

customAttribute *customstring*
Specifies for the value of *customstring* a custom attribute in string format as a Key Management Interoperability Protocol attribute. Client-specific attributes must start with the characters "x-" (x hyphen) and server-specific attributes must start with "y-" (y hyphen).

NAME
Client or server attribute name.

VALUE
Value of the attribute name.

link *objectname, objectnametarget*
Specifies the link from one managed cryptographic object to another, closely related target managed cryptographic object. This attribute is a Key Management Interoperability Protocol attribute.

TYPE
CERTIFICATE, PRIVATE_KEY, PUBLIC_KEY, DERIVATION_BASE_OBJECT, DERIVED_KEY, REPLACEMENT_OBJECT, REPLACED_OBJECT

LINKED_OBJECT_ID
Specify the target uuid of the linked object.

name

Specifies the name that is used to identify or locate the object. This attribute is a Key Management Interoperability Protocol attribute.

TYPE

TEXT, URI

VALUE

Name, or URI identifying the object.

objectGroup *objectgroupname1, objectgroupnameN*

Specifies for *objectgroupname1, objectgroupnameN* the values of one or more object group names of which this object might be part. This attribute is a Key Management Interoperability Protocol attribute.

VALUE

Object group name.

index

Specify the index to update or delete an attribute value.

operation

Required. Specify one of these valid operations to run on an attribute value: add, update, delete

-uuid

Required. Specify the Universal Unique Identifier of the certificate.

Example

This Jython-formatted command adds an attribute to a certificate.

```
print AdminTask.tklmCertAttributeUpdate
 ('[-uuid CERTIFICATE-d3ee4491-f96e-495d-bb37-fc03748924ba
  –operation add –attrName cryptoParams
   –attrValue "{MODE CBC} {PAD NONE} {HASH SHA256} {ROLE BDK}"]')
```

This Jython-formatted command adds an attribute for a certificate name to a certificate.

```
print AdminTask.tklmCertAttributeUpdate
 ('[-uuid CERTIFICATE-d3ee4491-f96e-495d-bb37-fc03748924ba
  –operation add –attrName name
  –attrValue "{TYPE TEXT} {VALUE cert name for xyz}"]')
```

This Jython-formatted command updates an attribute for a certificate name.

```
print AdminTask.tklmCertAttributeUpdate
 ('[-uuid KEY-d3ee4491-f96e-495d-bb37-fc03748924ba
  -operation update -index 0 -attrName name
   -attrValue "{TYPE TEXT} {VALUE updated cert name for xyz}"]')
```

This Jython-formatted command deletes the value at index 0 for the **attrName** attribute.

```
print AdminTask.tklmCertAttributeUpdate
 ('[-uuid KEY-d3ee4491-f96e-495d-bb37-fc03748924ba
  -operation delete -index 0 -attrName name]')
```

tklmCertCreate

Use the **tklmCertCreate** command to create a certificate and a public and private key pair, and store the certificate in an existing keystore.

Note: The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager. Use the REST interfaces instead.

Purpose

Use this command to create a certificate and a public and private key pair, and store the certificate in an existing keystore.

`Do not use other key-generating tools such as keytool or the iKeyman utility` to create or to modify keys or certificates. Use IBM Security Key Lifecycle Manager.

Note: If you additionally want to specify that a certificate is used as the:

- System default or partner certificate

  You must use the **`tklmDeviceGroupAttributeList`** and **`tklmDeviceGroupAttributeUpdate`** commands to view and change values for use as the system default or partner certificate. These values were previously stored in the obsolete **`drive.default.alias1`** (for system default) or **`drive.default.alias2`** (for system partner) properties.
- `SSLSERVER`

  Use the **`tklmConfigUpdateEntry`** command to update the value of the **`config.keystore.ssl.certalias`** property entry in the `SKLMConfig.properties` file.

Permissions

Your role must have a permission to the create action and permission to the appropriate device group. Or, your role must have a permission to the configure action to create an SSL or KMIP certificate. To make this certificate the default, your role must have permission to the modify action.

Syntax

tklmCertCreate -type *type* **-alias** *certalias* **-cn** *commonname* **-ou** *organizationunit* **-o** *organization* **-locality** *locality* **-state** *state* **-country** *country* **-keyStoreName** *keystorename* **-usage** {3592 | DS8000 | GENERIC | *userdevicegroup* | SSLSERVER | SSLCLIENT } **-validity** *integerindays*

Parameters

`-alias`
: Required. Specify a unique name for the certificate. The name is not case-sensitive. If you specify `MY Cert1`, the value is stored as `my cert1`.

  Note: Do not use a value such as `aaa000000000000000002` where the value begins with three alphabetic characters followed by 18 numeric characters. IBM Security Key Lifecycle Manager uses this format to generate a key group with symmetric keys.

  Do not use forward slash (/) or backslash (\) characters in the value.

`-cn`
: Required. Specify the common name.

-country
Specify a country as a two-letter country code.

-keyStoreName
Required. Specify the name of an existing keystore.

-locality
Specify a locality, such as a city.

-o Specify the organization. For example, `o=myCompanyName`.

-ou
Specify the organizational unit. For example, `ou=marketing`.

-state
Specify a state or province.

-type
Required. Specify a certificate type. You can specify the following certificate types:

- Self-signed

  The subject name and the issuer name of the certificate are the same. To create a request for a certificate that is not self-signed, use the **tklmCertGenRequest** command to create a user certificate in which the certificate issuer name represents a certificate authority, and the subject name represents a user or an end entity.

-usage
Required. Specify the target application usage, such as SSLSERVER. You can specify the following values:

3592 Specifies the 3592 device group.

DS8000
Specifies the DS8000 device group.

GENERIC
Specifies a device family that uses the Key Management Interoperability Protocol to interact with IBM Security Key Lifecycle Manager. The GENERIC device group enables management of KMIP objects.

Do not use the command-line interface to add a device to the GENERIC device group, or to change a GENERIC device group attribute.

SSLCLIENT
Client-side certificate that is used in secure communication by using Secure Socket Layer protocol to authenticate the client device.

SSLSERVER
Server-side certificate that is used in secure communication by using Secure Socket Layer protocol.

userdevicegroup
Specifies a user-defined group that is based on a supported device family.

-validity
Required. Specify the interval of time in days during which the certificate is valid. The interval can range from a minimum of one day to a maximum of 9000 days.

Example

This Jython-formatted command creates a self-signed certificate with an alias of `tklmCertificate` that is valid for 999 days.

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

This Jython-formatted command creates a certificate for SSL authentication.

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

This Jython-formatted command creates a IKEv2-SCSI certificate for authentication.

```
print AdminTask.tklmCertCreate ('[-type selfsigned
 -alias sklmIKEV2SCSICertificate -cn sklmikev2scs1 -ou accounting
  -o myCompanyName -country US -keyStoreName defaultKeyStore
    -usage IKEV2SERVER -validity 999]')
```

tklmCertDefaultRolloverAdd

Use the **`tklmCertDefaultRolloverAdd`** command to add a default certificate rollover for a specific date and device group. The rollover certificate takes the place of the previous default certificate.

Note: The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager. Use the REST interfaces instead.

Purpose

Use this command to add a default certificate rollover for a specific date and device group. You can specify that the certificate is used as the system default or the partner certificate, or both. The rollover certificate takes the place of the previous default certificate.

Permissions

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

Syntax

tklmCertDefaultRolloverAdd -usage {3592 | *userdevicegroup*} **-alias** *certalias* **-certDefaultType** *certdefault* **-effectiveDate** *effectivedatevalue*

Parameters

`-alias`
Required. Specify the name of the existing certificate.

`-certDefaultType`
Required. Specify whether the certificate is used as the system default or partner certificate, or both. You can specify the following values:

1 Certificate is the system default.

2 Certificate is the partner certificate.

3 Certificate is used as both the system default and the partner certificate.

-effectiveDate
Required. Specify the rollover date on which the certificate becomes the default system or partner certificate. The value is a current or future date in *yyyy-MM-dd* format.

-usage
Required. Specify the device group. You can specify the following values:

3592 Specifies the 3592 device group.

userdevicegroup
Specifies a new, user-defined instance of a supported 3592 device family.

Example

This Jython-formatted command specifies an existing, default certificate for 3592 tape drives.

```
print AdminTask.tklmCertDefaultRolloverAdd
('[-usage 3592 -alias tklmcert1
     -certDefaultType 1 -effectiveDate 2010-05-30]')
```

tklmCertDefaultRolloverDelete

Use the **tklmCertDefaultRolloverDelete** command to remove a certificate rollover that is specified in a rollover list.

Note: The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager. Use the REST interfaces instead.

Purpose

Use this command to remove a certificate rollover that is specified in a rollover list.

Permissions

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

Syntax

tklmCertDefaultRolloverDelete -uuid *certrolloveruuid*

Parameters

-uuid
Required. Specify the Universal Unique Identifier of an existing certificate rollover. Use the **tklmCertDefaultRolloverList()** command to list attributes, including the value of the **-uuid** parameter. For example: 101

Example

This Jython-formatted command removes a certificate rollover that is specified in a rollover list.

```
print AdminTask.tklmCertDefaultRolloverDelete
('[-uuid 101]')
```

tklmCertDefaultRolloverList

Use the **tklmCertDefaultRolloverList** command to list certificate rollovers in a rollover list for a device group.

Note: The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager. Use the REST interfaces instead.

Purpose

Use this command to list certificate rollovers in a rollover list for a device group.

Permissions

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

Syntax

tklmCertDefaultRolloverList -usage {3592 | *userdevicegroup*} {**-uuid** *certrolloveruuid* | **-name** *certalias*} **-v** {y | n}

Parameters

-name
Optional. Specify the alias of a certificate. For example: `my3592cert`

-usage
Required. Specify the device group. You can specify the following values:

3592 Specifies the 3592 device group.

userdevicegroup
Specifies a new, user-defined instance of a supported 3592 device family.

-uuid
Optional. Specify the unique universal identifier of an existing certificate rollover. For example: `101`

-v [y | n]
Optional. Verbose. The default is `n`, or no extra information. To list more information about a certificate, specify:

```
-v y
```

Example

This Jython-formatted command lists certificate rollovers that are available in a rollover list for 3592 tape drives.

```
print AdminTask.tklmCertDefaultRolloverList
('[-usage 3592]')
```

tklmCertDelete

Use the **tklmCertDelete** command to delete a 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 default or partner. You cannot delete a certificate that is scheduled for a future rollover. You also cannot delete the active SSLSERVER or IKEV2SERVER certificate.

Note: The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager. Use the REST interfaces instead.

Purpose

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.

Use this command to delete a 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 default or partner. You cannot delete a certificate that is scheduled for a future rollover. You also cannot delete the active SSLSERVER certificate.

Deleting a certificate deletes the material from the database.

Permissions

Your role must have a permission to the delete action and a permission to the appropriate device group. Or, your role must have a permission to the configure action to delete an SSL or KMIP certificate.

Syntax

tklmCertDelete -alias *certalias* **-keyStoreName** *keystorename*

Parameters

-alias
Required. Specify a unique name for the certificate.

-keyStoreName
Required. Specify the name of an existing keystore.

Example

This Jython-formatted command deletes a certificate that is not currently associated with a device.

```
print AdminTask.tklmCertDelete ('[-alias mycertalias
 -keyStoreName defaultKeyStore]')
```

tklmCertExport

Use the **tklmCertExport** command to export a certificate file.

Note: The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager. Use the REST interfaces instead.

Purpose

Use this command to export a certificate file.

Permissions

Your role must have a permission to the get action and a permission to the appropriate device group. Or, your role must have a permission to the configure action to export an SSL or KMIP certificate.

Syntax

tklmCertExport -uuid *universalCertID* **-fileName** *pathandfilename* **-format** {base64 | DER}

Parameters

-fileName
Required. Specify the path and file name to which the certificate is exported.

If you specify no path, or a relative path, the command appends the file and the path, if specified, to the *SKLM_HOME* directory. If you specify an absolute path, the file is stored in that path, which is *not* relative to the *SKLM_HOME* directory.

-format
Specify either base64 (default, if this parameter is not specified) or DER (Distinguished Encoding Rules) format.

-uuid
Required. Specify the Universal Unique Identifier of the certificate.

Example

This Jython-formatted command exports a certificate.

```
print AdminTask.tklmCertExport
 ('[-uuid CERTIFICATE-61f8e7ca-62aa-47d5-a915—8adbfbdca9de
  -format DER -fileName d:\\mypath\\mycertfilename.der]')
```

tklmCertGenRequest

Use the **tklmCertGenRequest** command to create a PKCS #10 certificate request file.

Note: The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager. Use the REST interfaces instead.

Purpose

Use this command to create a PKCS #10 certificate request file.

The return value from the command is the full path name of the generated certificate request file, such as *SKLM_HOME*\080419154137—sslcert001.csr. You must manually send the request to a certificate authority.

When the certificate authority returns a certificate in response to this request, copy the certificate to a file. Use the **tklmCertImport** command to load the response file. You must specify the same alias name that was used with the **tklmCertGenRequest** command to generate the request.

After you generate the certificate request, the certificate activation date and creation date are identical and this certificate is available to the key server and drive.

Permissions

Your role must have a permission to the create action and a permission to the appropriate device group. Or, your role must have a permission to the configure action to create a certificate request for an SSL or, KMIP certificate.

Syntax

tklmCertGenRequest -alias *certalias* **-cn** *commonname* **-ou** *organizationunit* **-o** *organization* **-locality** *locality* **-state** *state* **-country** *country* **-keyStoreName** *keystorename* **-fileName** *certfilename* **-usage** {3592 | DS8000 | GENERIC | *userdevicegroup* | SSLSERVER | SSLCLIENT } **-validity** *integerindays*

Parameters

-alias
Required. Specify a unique name for the certificate. Retain a record of the alias value of the certificate request, for use when you import the returned certificate.

-cn
Required. Specify the common name.

-country
Specify a country as a two-letter country code.

-fileName
Required. Specify the name of the certificate request file, which is created on the IBM Security Key Lifecycle Manager server, relative to the *SKLM_HOME* directory.

SKLM_HOME is the base directory that contains the IBM Security Key Lifecycle Manager code and configuration.

Distributed systems

- Windows systems: *drive*:\Program Files (x86)\IBM\WebSphere\AppServer\products\sklm
- Systems such as AIX or Linux: *path*/IBM/WebSphere/AppServer/products/sklm

This location is also stored in the sklm.properties file that is in the *WAS_HOME*\properties directory.

-keyStoreName
Required. Specify the name of the keystore.

-locality
Specify a locality, such as a city.

-o Specify the organization. For example, o=myCompanyName.

-ou
Specify the organizational unit. For example, ou=marketing.

-state
Specify the full name of a state or province.

-usage

Required. Specify the target application usage, such as SSLSERVER. You can specify the following values:

3592 Specifies the 3592 device group.

DS8000
Specifies the DS8000 device group.

GENERIC
Specifies a device family that uses the Key Management Interoperability Protocol to interact with IBM Security Key Lifecycle Manager. The GENERIC device group enables management of KMIP objects.

Do not use the command-line interface to add a device to the GENERIC device group, or to change a GENERIC device group attribute.

SSLCLIENT
Client-side certificate that is used in secure communication by using Secure Socket Layer protocol to authenticate the client device.

SSLSERVER
Server-side certificate that is used in secure communication by using Secure Socket Layer protocol.

userdevicegroup
Specifies a user-defined group that is based on a supported device family.

-validity

Required. Specify the interval of time in days during which the certificate is valid. The interval can range from a minimum of one day to a maximum of 9000 days.

Examples

These Jython-formatted commands create requests for new certificates that are valid for 999 days.

- SSL communication

```
print AdminTask.tklmCertGenRequest('[-alias sklmSSLCertificate1
 -cn sklm -ou sales -o myCompanyName -locality myLocation
  -country US -validity 999 -keyStoreName defaultKeyStore
   -fileName mySSLCertRequest1.crt -usage SSLSERVER]')
```

- 3592 tape drives

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

- DS8000 Turbo drives

```
print AdminTask.tklmCertGenRequest('[-alias sklmCertificate3
 -cn sklm -ou sales -o myCompanyName -locality myLocation
  -country US -validity 999 -keyStoreName defaultKeyStore
   -fileName myCertRequest3.crt -usage DS8000]')
```

tklmCertImport

Use the **tklmCertImport** command to import a certificate.

Note: The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager. Use the REST interfaces instead.

Purpose

Use this command to import a certificate file.

If the specified alias exists in the specified keystore, then the specified file name must contain a certificate response that was generated by a certificate authority in response to a certificate request that the **tklmCertGenRequest** command generates. The specified alias must have been created by using a **tklmCertGenRequest** command. The subject DN fields in the specified file must match the values that are specified in the request that the **tklmCertGenRequest** command generates. A CA-issued certificate that is stored in IBM Security Key Lifecycle Manager cannot have a string representation of the integer value of the serial number greater than 64 characters.

If the specified alias does not exist in the specified keystore, then the specified file name must contain a certificate to load into the keystore.

You cannot import certificates that exceed 2048 bits in length. If you encounter this problem, you might want to generate a certificate that has a key length less than or equal to 2048 bits, or use alternate certificates.

Note: If you additionally want to specify that a certificate is used as the:

- System default or partner certificate

  You must use the **tklmDeviceGroupAttributeList** and **tklmDeviceGroupAttributeUpdate** commands to view and change values for use as the system default or partner certificate. These values were previously stored in the obsolete **drive.default.alias1** (for system default) or **drive.default.alias2** (for system partner) properties.
- SSLSERVER

  Use the **tklmConfigUpdateEntry** command to update the value of the **config.keystore.ssl.certalias** property entry in the SKLMConfig.properties file.

Permissions

Your role must have a permission to the create action and a permission to the appropriate device group. Or, your role must have a permission to the configure action to import an SSL or KMIP, certificate.

Syntax

tklmCertImport -filename *certfilename* **-format** {base64 | DER} **-alias** *certalias* **-keyStoreName** *keystorename* **-usage** {3592 | DS8000 | GENERIC | *userdevicegroup* | SSLSERVER | SSLCLIENT }

Parameters

-alias
  Required. Specify a unique name for the certificate.

-fileName
  Required. Specify the file name to import containing the certificate data. The

imported file is stored in IBM Security Key Lifecycle Manager in a keystore location relative to the *SKLM_HOME* directory.

-format
Specify either base64 (default) or DER (Distinguished Encoding Rules) format.

-keyStoreName
Required. Specify the name of the keystore.

-usage
Required. Specify the target application usage, such as SSLSERVER. You can specify the following values:

3592 Specifies the 3592 device group.

DS8000
Specifies the DS8000 device group.

GENERIC
Specifies a device family that uses the Key Management Interoperability Protocol to interact with IBM Security Key Lifecycle Manager. The GENERIC device group enables management of KMIP objects.

Do not use the command-line interface to add a device to the GENERIC device group, or to change a GENERIC device group attribute.

SSLCLIENT
Client-side certificate that is used in secure communication by using Secure Socket Layer protocol to authenticate the client device.

SSLSERVER
Server-side certificate that is used in secure communication by using Secure Socket Layer protocol.

SYSLOG
Syslog server-side certificate that is used in secure communication by using Secure Socket Layer protocol to authenticate the syslog server.

userdevicegroup
Specifies a user-defined group that is based on a supported device family.

Example

These Jython-formatted commands import certificates.

- SSL communication

```
print AdminTask.tklmCertImport
 ('[-fileName myTempPath\\mySSLCertRequest1.cer
   -alias sklmSSLCertificate1 -format base64
     -keyStoreName defaultKeyStore -usage SSLSERVER]')
```

- 3592 tape drives

```
print AdminTask.tklmCertImport \
 ('[-fileName myTempPath\\myCertRequest2.cer
   -alias sklmCertificate2 -format base64
     -keyStoreName defaultKeyStore -usage 3592]')
```

- DS8000 Turbo drives

```
print AdminTask.tklmCertImport
 ('[-fileName myTempPath\\myCertRequest3.cer
   -alias sklmCertificate3 -format base64
     -keyStoreName defaultKeyStore -usage DS8000]')
```

tklmCertList

Use the **tklmCertList** command to return certificate information, which is based on criteria such as a specific state.

Note: The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager. Use the REST interfaces instead.

Purpose

Use this command to return certificate information, which is based on specified criteria such as a specific state.

You can list a specific certificate by specifying:

- uuid
- alias and keystore name

Alternatively, you can list all certificates in a specified state. If you do not specify any parameter, you can list all certificates.

Note:

- The KMIP interface for IBM Security Key Lifecycle Manager, version 2.5 does not support the certificate object. IBM Security Key Lifecycle Manager creates certificate objects internally, but does not set some KMIP required and optional attributes.

  Running the **tklmCertList** command in verbose mode lists many of the KMIP attributes as NULL. The null values do not affect the IBM Security Key Lifecycle Manager function.
- Use the **tklmCertList** command to find certificates that are marked as `CONFLICTED` or `UNKNOWN`. Specify no value for the **-usage** parameter, or specify a parameter value of 3592, DS8000, or `SSLSERVER`. For example, this Jython-formatted command lists all certificates for the 3592 device group:

  ```
  print AdminTask.tklmCertList('[-usage 3592 -v y]')
  ```
- The **tklmCertList** command returns only the first 2000 certificates.

Permissions

Your role must have a permission to the view action and a permission to the appropriate device group. Or, your role must have a permission to the configure action to view an SSL or KMIP certificate.

Syntax

tklmCertList -uuid *universalCertID* **-alias** *certalias* **-keyStoreName** *keystorename* **-attributes** [state *value*] **-usage** {3592 | DS8000 | GENERIC | *userdevicegroup* | SSLSERVER | SSLCLIENT } **-v** {y | n}

Parameters

There are no required parameters.

-alias
: Specify a unique name for the certificate.

-attributes
Specify the attributes to search for. *Only the state attribute and the trusted attribute are supported; only one state can be specified in a command instance.*

state You can include the following values for the state attribute:

pending
A certificate request entry is pending the return of a certificate that is approved and certified by a certificate authority.

pre-active
Object exists but is not yet usable for any cryptographic purpose, such as migrated certificates with a future use time stamp.

active
Object is in operational use for protecting and processing data that might use **Process Start Date** and **Protect Stop Date** attributes. For example, protecting includes encryption and signature issue. Processing includes decryption and signature verification.

compromised
The security of the object is suspect for some reason. A compromised object never returns to an `uncompromised` state, and cannot be used to protect data. Use the object only to process cryptographically protected information in a client that is trusted to handle compromised cryptographic objects.

IBM Security Key Lifecycle Manager retains the state of the object immediately before it was compromised. To process data that was previously protected, the compromised object might continue to be used.

deactivated
Object is not to be used to apply cryptographic protection such as encryption or signing. However, if extraordinary circumstances occur, the object can be used with special permission to process cryptographically protected information. For example, processing includes decryption or verification.

destroyed
Object is no longer usable for any purpose. This status causes the object to be removed from the product.

destroyed-compromised
Object is no longer usable for any purpose. This status causes the object to be removed from the product.

trusted
Values are y, n, or no value.

Set the value to y to list only trusted certificates. Set this value to n to list only untrusted certificates. Not setting a value lists both trusted and untrusted certificates.

-keyStoreName
Specify the name of the keystore.

-usage
Specify the target application usage, such as SSLSERVER. You can specify the following values:

3592 Specifies the 3592 device group.

DS8000
Specifies the DS8000 device group.

GENERIC
Specifies a device family that uses the Key Management Interoperability Protocol to interact with IBM Security Key Lifecycle Manager. The GENERIC device group enables management of KMIP objects.

Do not use the command-line interface to add a device to the GENERIC device group, or to change a GENERIC device group attribute.

SSLCLIENT
Client-side certificate that is used in secure communication by using Secure Socket Layer protocol to authenticate the client device.

SSLSERVER
Server-side certificate that is used in secure communication by using Secure Socket Layer protocol.

SYSLOG
Syslog server-side certificate that is used in secure communication by using Secure Socket Layer protocol to authenticate the syslog server.

userdevicegroup
Specifies a user-defined group that is based on a supported device family.

-uuid
Specify the Universal Unique Identifier of the certificate. For example, `CERTIFICATE-b4c70958-446d-42c4-ae3b-8c9e0f44c0fa` might be the value.

-v [y | n]
Verbose. The default is `n`, or no extra information. To list more information about a certificate, specify `y` (for yes):

```
-v y
```

Example

This Jython-formatted command lists certificates that are in active state.

```
print AdminTask.tklmCertList('[-usage 3592
 -attributes "{state active}" -v y]')
```

This command lists the first 2000 certificates.

```
print AdminTask.tklmCertList()
```

tklmCertUpdate

Use the **tklmCertUpdate** command to update attributes or usage for a certificate. For example, you might update the state of the certificate to indicate that its use is compromised. If a certificate is not associated with a device or scheduled for a future rollover, you might also change the usage of the certificate from one device group to a related device group in the same device family. For example, you might change usage from 3592 to a user-defined device group such as `my3592`.

Note: The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager. Use the REST interfaces instead.

Purpose

Use this command to update attributes or usage for a certificate. For example, you might update the state of the certificate to indicate that its use is compromised. If a certificate is not associated with a device or scheduled for a future rollover, you might also change the usage of the certificate from one device group to a related device group in the same device family. For example, you might change usage from 3592 to a user-defined device group such as `my3592`.

Permissions

Your role must have a permission to the modify action and a permission to the appropriate device group. Or, your role must have a permission to the configure action to modify an SSL or KMIP certificate.

Syntax

tklmCertUpdate -uuid *universalCertID* **-usage** {3592 | DS8000 | GENERIC | *userdevicegroup* | SSLSERVER | SSLCLIENT } **-attributes** {*attributevaluepair*} {*attributevaluepair*}

Parameters

`-attributes`

Specify one or more of these attribute-value pairs:

Note: Do not use an asterisk (*) or question mark (?) as a character in a Key Management Interoperability Protocol attribute. These wildcard characters are reserved for future use.

`compromised`

Specifies whether the use is `compromised`. The only value is y (`compromised`). You cannot change a `compromised` key or certificate to an `uncompromised` state.

`information` *`informationstring`*

Specifies more information about the use of an object.

`trusted [y|n]`

Specifies whether the use is trusted. Set this value to `y` to mark the key or certificate as trusted, or set a value of `n` to mark the key or certificate as not trusted. You cannot set compromised or expired keys or certificates to be trusted.

`-usage`

Specify the target application usage, such as `SSLSERVER`. You can specify the following values:

3592 Specifies the 3592 device group.

DS8000

Specifies the DS8000 device group.

GENERIC

Specifies a device family that uses the Key Management Interoperability Protocol to interact with IBM Security Key Lifecycle Manager. The GENERIC device group enables management of KMIP objects.

Do not use the command-line interface to add a device to the GENERIC device group, or to change a GENERIC device group attribute.

SSLCLIENT
Client-side certificate that is used in secure communication by using Secure Socket Layer protocol to authenticate the client device.

SSLSERVER
Server-side certificate that is used in secure communication by using Secure Socket Layer protocol.

userdevicegroup
Specifies a user-defined group that is based on a supported device family.

-uuid
Required. Specify the Universal Unique Identifier of the certificate.

Example

This Jython-formatted command updates the state of a certificate to show that its use is compromised.

```
print AdminTask.tklmCertUpdate
 ('[-uuid CERTIFICATE-28559cfc-9031-46e9-a830-159bfcbe0d23
  -attributes "{compromised y}"]')
```

This Jython-formatted command updates the value of the `information` attribute that contains two words for a certificate in the IBM Security Key Lifecycle Manager database.

```
print AdminTask.tklmCertUpdate
 ('[-uuid CERTIFICATE-33fc26e-5fb5a0e66143
   -usage DS8000 -attributes "{information {new information}}"]')
```

This Jython-formatted command updates an IKEv2-SCSI certificate to show that it is trusted.

```
print AdminTask.tklmCertUpdate
 ('[-uuid CERTIFICATE-44337cfc-6036-46e9-a830-555bfcbe0d23
  -attributes "{trusted y}"]')
```

tklmConfigDeleteEntry

Use the **tklmConfigDeleteEntry** command to delete a property in the IBM Security Key Lifecycle Manager configuration file. For example, you might delete a customized property.

Note: The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager. Use the REST interfaces instead.

Purpose

Deletes a property in the `SKLMConfig.properties` file, which is the IBM Security Key Lifecycle Manager configuration file. Required properties cannot be deleted, and cause an error.

The return message from this command indicates whether the change occurs immediately, or you must restart the IBM Security Key Lifecycle Manager server for the change to take effect.

Permissions

Your role must have a permission to the configure action.

Syntax

tklmConfigDeleteEntry -name *attributename*

Parameters

-name
Required. Specify the attribute.

Example

This Jython-formatted command deletes a customized property in the configuration file.

```
print AdminTask.tklmConfigDeleteEntry ('[-name myCustomizedAttribute]')
```

tklmConfigGetEntry

Use the **tklmConfigGetEntry** command to return the current value or values of a property in the IBM Security Key Lifecycle Manager configuration file.

Note: The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager. Use the REST interfaces instead.

Purpose

Gets the current value or values of a property in the SKLMConfig.properties file.

To discover which property names currently exist, use the **tklmConfigList** command. If the specified name does not exist in the property file, the command returns a message to indicate that the property name does not exist.

Permissions

Your role must have a permission to the configure action.

Syntax

tklmConfigGetEntry -name *propertyname*

Parameters

-name
Required. Specify the name of the property in the SKLMConfig.properties configuration file.

Examples

This Jython-formatted command obtains the current value of the **Audit.event.types**property in the SKLMConfig.properties configuration file.

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

This Jython-formatted command obtains the current value of the **TransportListener.tcp.port** property in the SKLMConfig.properties configuration file.

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

tklmConfigList

Use the **tklmConfigList** command to list the contents of the IBM Security Key Lifecycle Manager configuration file.

Note: The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager. Use the REST interfaces instead.

Purpose

List the contents of the SKLMConfig.properties file, which is the IBM Security Key Lifecycle Manager configuration file.

Permissions

You must have a valid role, such as klmSecurityOfficer, or a valid permission to an action such as Configure, or a valid permission to an action and a device group.

Syntax

tklmConfigList -v {y| n}

Parameters

There are no required parameters.

-v [y | n]
Verbose. The default is n. This command returns verbose information, regardless of the verbose setting.

Example

This Jython-formatted command verbosely lists the contents of the IBM Security Key Lifecycle Manager configuration file.

```
print AdminTask.tklmConfigList()
```

tklmConfigUpdateEntry

Use the **tklmConfigUpdateEntry** command to change an existing entry or to add an entry in the SKLMConfig.properties file.

Note: The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager. Use the REST interfaces instead.

Purpose

Changes an existing entry or adds an entry in the SKLMConfig.properties file. Use the **tklmDeviceGroupAttributeUpdate** command to change an attribute of a device group in the IBM Security Key Lifecycle Manager database.

Note: If you use the graphical user interface, or command-line interface, you can change the IBM Security Key Lifecycle Manager configuration file while the server is running. Depending on the change, you might see a message that indicates you to restart the IBM Security Key Lifecycle Manager server for the change to take effect.

Permissions

Your role must have a permission to the configure action.

Syntax

tklmConfigUpdateEntry -name *attributename* **-value** *attributevalue*

Parameters

-name
Required. Specify the name of the attribute.

-value
Required. Specify the value of an attribute. Depending on the attribute, you can specify multiple values, by using commas to separate the values. For more information, see the `SKLMConfig.properties` file.

Examples

This Jython-formatted command example changes the types of events that are audited, specifying an Audit.event.types property to have two values (`runtime` and `audit_management`) in the `SKLMConfig.properties` file.

```
print AdminTask.tklmConfigUpdateEntry ('[-name Audit.event.types
 -value runtime,audit_management]')
```

This example specifies a different TCP port number.

```
print AdminTask.tklmConfigUpdateEntry
 ('[-name TransportListener.tcp.port -value 3802]')
```

tklmDeviceAdd

Use the **`tklmDeviceAdd`** command to add a device to the IBM Security Key Lifecycle Manager database. If the device is a DS5000 storage server, the command can optionally create a machine-to-device relationship.

Note: The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager. Use the REST interfaces instead.

Purpose

Use this command to add a device to the IBM Security Key Lifecycle Manager database. If the device is a DS5000 storage server, the command can create a machine-to-device relationship.

Depending on whether IBM Security Key Lifecycle Manager is configured to allow unknown devices, this command can be used to limit which devices can access the server. It can also be used to specify which set of keys are used for requests for a particular device. If IBM Security Key Lifecycle Manager receives a request from a device, which does not exist in the IBM Security Key Lifecycle Manager database and it is not configured to allow unknown devices, the request is rejected.

To specify that IBM Security Key Lifecycle Manager automatically serves keys upon request, configure the **device.AutoPendingAutoDiscovery** attribute, which provides auto pending and auto discovery functions.

Permissions

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

Syntax

tklmDeviceAdd -type {LTO | 3592 | DS5000 | DS8000 | BRCD_ENCRYPTOR | ONESECURE | GENERIC | *userdevicegroup*} **-serialNumber** *devicenumber* **-attributes** {*attributevaluepair*} {*attributevaluepair*}

Parameters

-attributes

Specify one or more attribute-value pairs. Their values are stored in the IBM Security Key Lifecycle Manager database.

You can specify the following device attributes:

aliasOne

Specifies a default alias for a certificate that is used by a 3592 tape drive or a DS8000 Turbo drive. Not used for an LTO tape drive or DS5000 storage server.

- 3592 tape drive

  The value is optional for a 3592 tape drive and specifies the primary certificate that the device in the 3592 device family uses if the primary certificate is not available. If this attribute is not specified, the partner default certificate is used, as specified by a table entry for the device group in the IBM Security Key Lifecycle Manager database.

- DS8000 Turbo drive

  The value is optional for a DS8000 Turbo drive and matches the label **Primary certificate for image** in the graphical user interface panels for a DS8000 Turbo drive.

Use the **tklmDeviceGroupAttributeList** and **tklmDeviceGroupAttributeUpdate** commands to view and change the value. This value was previously stored in the obsolete configuration parameter **drive.default.alias1**.

aliasTwo

Used for a 3592 tape drive or a DS8000 Turbo drive. Not used for an LTO tape drive or DS5000 storage server.

- 3592 tape drive

  This attribute specifies a default alternative alias for a 3592 tape drive. This value can be the same, or different from the value that is specified for the primary certificate.

  The value specifies the secondary certificate that the device in the 3592 device family uses if the primary certificate is not available. If this attribute is not specified, the partner default certificate is used, as specified by a table entry for the device group in the IBM Security Key Lifecycle Manager database.

- DS8000 Turbo drive

  For a device in the DS8000 device family, the value specifies a secondary certificate that is available for use. For example, you might use this certificate to unlock a DS8000 Turbo drive in the case of a deadlock condition.

Use the **tklmDeviceGroupAttributeList** and **tklmDeviceGroupAttributeUpdate** commands to view and change the value. This value was previously stored in the obsolete configuration parameter **drive.default.alias2**.

description
Specifies more information that describes the type of device or its purpose.

deviceText
Optional. Specifies unique text with a minimum length greater than zero bytes and a maximum length of 96 bytes that describes a DS5000 storage server. Use the **tklmDeviceUpdate** command to update this value.

driveCert
Specifies the actual certificate that is used to identify the device (in base64 encoded format). For current devices, this field is not in use.

keyPrefix
Specifies a key prefix as part of the key name. To add new keys, specify the prefix and number of keys. This value is used only for a device in the DS5000 device family.

machineID
Optional unless you want to add the association of a device to an existing machine identifier in the IBM Security Key Lifecycle Manager database. Specifies a unique machine ID for a DS5000 storage server, which is a concatenation of the worldwide name and the volume serial number. For example, specify 304238303030343700000000000. Use the **tklmMachineIdentityList** command to locate machine identities. Use the **tklmMachineDeviceDelete** command to remove the association of a device from an existing machine identifier.

numberOfKeys
Specifies the number of keys to generate. The keys use the value of the `keyPrefix` attribute. The maximum number of keys is 12. If this value is not specified, the default value is 12 keys.

This value is used only for a device in the DS5000 device family.

symAlias
Specifies an alias that is used to identify an existing key group for an LTO tape drive. The attribute is also used for DS5000 storage server to change or associate a new device key container.

The value of symAlias is used to specify which symmetric key group is used to obtain a key for a new device media instance. If this attribute is not specified, then the value of the symmetricKeySet attribute is used.

For compatibility with an earlier version of the Encryption Key Manager product, you can also specify the alias of an existing key entry.

worldwideName
Specifies the name of a device, which is a nonsecure address that is used in combination with other device information, such as a serial number, to define devices and device paths. Specify a 16-character hexadecimal value that contains only the characters `ABCDEFabcdef1234567890`.

`-serialNumber`
Required. Specify the serial number as an ASCII string. The value is case-sensitive. You can use alphanumeric characters and the following special characters: periods, spaces, dashes, semicolons, and underscore. Do not use a space at the beginning or end of a serial number.

- LTO tape drives

  The serial number must be exactly 10, 12, or 24 characters in length. IBM Security Key Lifecycle Manager pads a serial number that is 10 characters in length with two leading zeros.
- 3592 tape drives and DS8000 Turbo drives

  The case-sensitive value must be exactly 12 characters in length.
- DS5000 storage servers

  The serial number can vary between 1 and 48 characters in length. No padding occurs.

`-type`
Required. Specify the device group:

LTO Specifies the LTO device group.

3592 Specifies the 3592 device group.

DS5000
Specifies the DS5000 device group.

DS8000
Specifies the DS8000 device group.

BRCD_ENCRYPTOR
Specifies the BRCD_ENCRYPTOR device group that is in the LTO device family.

ONESECURE
Specifies the ONESECURE device group that is in the DS5000 device family.

GENERIC
Specifies a device family that uses the Key Management Interoperability Protocol to interact with IBM Security Key Lifecycle Manager. The GENERIC device group enables management of KMIP objects.

Do not use the command-line interface to add a device to the GENERIC device group, or to change a GENERIC device group attribute.

userdevicegroup
Specifies a user-defined group that is based on a supported device family.

Example

This Jython-formatted command adds an LTO tape drive to the IBM Security Key Lifecycle Manager database.

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

This Jython-formatted command adds a 3592 tape drive to the IBM Security Key Lifecycle Manager database.

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

This Jython-formatted command adds a DS8000 Turbo drive to the IBM Security Key Lifecycle Manager database.

```
print AdminTask.tklmDeviceAdd ('[-type DS8000 -serialNumber CCCB31403AFF
  -attributes "{worldwideName ABCdeF1234567890}
  {description salesDivisionDrive}
  {aliasOne myimagecertificate}"]')
```

This Jython-formatted command adds a DS5000 storage server to the IBM Security Key Lifecycle Manager database. The command adds 10 keys and machine association to an existing machine.

```
print AdminTask.tklmDeviceAdd ('[-type DS5000 -serialNumber CDA39403AQJF
  -attributes "{worldwideName ABCdeF1234567890}
  {description marketingDivisionDrive}
  {keyPrefix AEF}
  {numberOfKeys 10}
  {deviceText abcdefghijklmnopqrst}
  {machineID 3042383030303437000000000000}"]')
```

tklmDeviceDelete

Use the **`tklmDeviceDelete`** command to remove information that identifies a device from the IBM Security Key Lifecycle Manager database.

Note: The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager. Use the REST interfaces instead.

Purpose

Use this command to remove information that identifies a device from the IBM Security Key Lifecycle Manager database. Delete any association between a DS5000 device and machine before you delete the device.

Permissions

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

Syntax

tklmDeviceDelete -uuid *deviceID*

Parameters

-uuid
Required. Specify the Universal Unique Identifier of the device. For example, `DEVICE-74386920-148c-47b2-a1e2-d19194b315cf` might be the value.

Example

This Jython-formatted command deletes metadata, such as the device serial number, from the IBM Security Key Lifecycle Manager database.

```
print AdminTask.tklmDeviceDelete
 ('[-uuid DEVICE-74386920-148c-47b2-a1e2-d19194b315cf]')
```

tklmDeviceList

Use the **tklmDeviceList** command to list information about all devices of a device group, or a specific device in the IBM Security Key Lifecycle Manager database.

Note: The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager. Use the REST interfaces instead.

Purpose

Use this command to list information about all devices of a device group, or a specific device in the IBM Security Key Lifecycle Manager database.

Permissions

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

Syntax

tklmDeviceList **-type** {LTO | 3592 | DS5000 | DS8000 | BRCD_ENCRYPTOR | ONESECURE | GENERIC | *userdevicegroup*} **-uuid** *deviceID* **-v** {y | n}

Parameters

There are no required parameters.

-type
Specify the device group. The default is all device groups.

LTO Specifies the LTO device group.

3592 Specifies the 3592 device group.

DS5000
Specifies the DS5000 device group.

DS8000
Specifies the DS8000 device group.

BRCD_ENCRYPTOR
Specifies the BRCD_ENCRYPTOR device group that is in the LTO device family.

ONESECURE

Specifies the ONESECURE device group that is in the DS5000 device family.

GENERIC

Specifies a device family that uses the Key Management Interoperability Protocol to interact with IBM Security Key Lifecycle Manager. The GENERIC device group enables management of KMIP objects.

Do not use the command-line interface to add a device to the GENERIC device group, or to change a GENERIC device group attribute.

userdevicegroup

Specifies a user-defined group that is based on a supported device family.

`-uuid`

Specify the Universal Unique Identifier of the device. For example, `DEVICE-74386920-148c-47b2-a1e2-d19194b315cf` might be the value.

`-v [y | n]`

Verbose. The default is `n`, or no extra information. To list more information about a device, specify `y` (for yes).

Example

This Jython-formatted command lists information for all 3592 tape drives in the IBM Security Key Lifecycle Manager database.

```
print AdminTask.tklmDeviceList ('[-type 3592]')
```

tklmDeviceUpdate

Use the **`tklmDeviceUpdate`** command to update the attributes of a device in the IBM Security Key Lifecycle Manager database. If the attribute does not exist, it is added to the device entry.

Note: The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager. Use the REST interfaces instead.

Purpose

Use this command to update the attributes of a device in the IBM Security Key Lifecycle Manager database. If the attribute does not exist, it is added to the device entry.

Permissions

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

Syntax

tklmDeviceUpdate -type {LTO | 3592 | DS5000 | DS8000 | BRCD_ENCRYPTOR | ONESECURE | GENERIC | *userdevicegroup*} **-uuid** *deviceID* **-attributes** {*attributevaluepair*} {*attributevaluepair*}

Parameters

-uuid

Required. Specify the Universal Unique Identifier of the device. For example, `DEVICE-74386920-148c-47b2-a1e2-d19194b315cf` might be the value.

-type

Specify the device group.

LTO Specifies the LTO device group.

3592 Specifies the 3592 device group.

DS5000
Specifies the DS5000 device group.

DS8000
Specifies the DS8000 device group.

BRCD_ENCRYPTOR
Specifies the BRCD_ENCRYPTOR device group that is in the LTO device family.

ONESECURE
Specifies the ONESECURE device group that is in the DS5000 device family.

GENERIC
Specifies a device family that uses the Key Management Interoperability Protocol to interact with IBM Security Key Lifecycle Manager. The GENERIC device group enables management of KMIP objects.

Do not use the command-line interface to add a device to the GENERIC device group, or to change a GENERIC device group attribute.

userdevicegroup
Specifies a user-defined group that is based on a supported device family.

-attributes

Specify one or more attribute-value pairs.

You can specify the following device attributes:

aliasOne
Specifies a default alias for a certificate that is used by a 3592 tape drive or a DS8000 Turbo drive. Not used for an LTO tape drive or DS5000 storage server.

- 3592 tape drive

  The value is optional for a 3592 tape drive and specifies the primary certificate that the device in the 3592 device family uses if the primary certificate is not available. If this attribute is not specified, the partner default certificate is used, as specified by a table entry for the device group in the IBM Security Key Lifecycle Manager database.

- DS8000 Turbo drive

  The value is optional for a DS8000 Turbo drive and matches the label **Primary certificate for image** in the graphical user interface panels for a DS8000 Turbo drive.

Use the **tklmDeviceGroupAttributeList** and **tklmDeviceGroupAttributeUpdate** commands to view and change the value. This value was previously stored in the obsolete configuration parameter **drive.default.alias1**.

aliasTwo

Used for a 3592 tape drive or a DS8000 Turbo drive. Not used for an LTO tape drive or DS5000 storage server.

- 3592 tape drive

  This attribute specifies a default alternative alias for a 3592 tape drive. This value can be the same, or different from the value that is specified for the primary certificate.

  The value specifies the secondary certificate that the device in the 3592 device family uses if the primary certificate is not available. If this attribute is not specified, the partner default certificate is used, as specified by a table entry for the device group in the IBM Security Key Lifecycle Manager database.

- DS8000 Turbo drive

  For a device in the DS8000 device family, the value specifies a secondary certificate that is available for use. For example, you might use this certificate to unlock a DS8000 Turbo drive in the case of a deadlock condition.

Use the **tklmDeviceGroupAttributeList** and **tklmDeviceGroupAttributeUpdate** commands to view and change the value. This value was previously stored in the obsolete configuration parameter **drive.default.alias2**.

description

Specifies more information that describes the type of drive or its purpose.

deviceText

Specifies unique text not greater than 96 bytes in length that describes a DS5000 storage server.

serialNumber

For a DS5000 storage server, specifies the serial number of drive. You can change the serial number of a DS5000 storage server to another serial number that is not currently in use.

symAlias

Specifies an alias that is used to identify an existing key or key group for an LTO tape drive that you create. The attribute is also used for DS5000 storage server to change or associate a new device key container. This value is stored in the IBM Security Key Lifecycle Manager database.

worldwideName

Specifies the name of a device, which is a nonsecure address that is used in combination with other device information, such as a serial number, to define devices and device paths. Specify a 16-character hexadecimal value that contains only the characters `ABCDEFabcdef1234567890`.

Examples

This Jython-formatted command updates the value of the aliasTwo attribute of a 3592 tape drive in the IBM Security Key Lifecycle Manager database.

```
print AdminTask.tklmDeviceUpdate
 ('[-uuid DEVICE-64c588ad-5ed8-4934-8c84-64cb9e11d990
  -attributes "{aliasTwo myPartner99}"]')
```

This Jython-formatted command updates the value of the symAlias attribute of an LTO tape drive in the IBM Security Key Lifecycle Manager database.

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

This Jython-formatted command updates the value of the description attribute of a DS8000 Turbo drive in the IBM Security Key Lifecycle Manager database.

```
print AdminTask.tklmDeviceUpdate
 ('[-uuid DEVICE-15d499ad-3ad8-3333-8c84-64cb9e11d990
  -attributes "{description myDevice}"]')
```

tklmDeviceGroupAttributeDelete

Use the **`tklmDeviceGroupAttributeDelete`** command to delete an attribute of a device group such as `myLTO`.

Note: The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager. Use the REST interfaces instead.

Purpose

Use this command to delete an attribute of a device group such as `myLTO`.

Permissions

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

Syntax

tklmDeviceGroupAttributeDelete [-name {LTO | 3592 | DS5000 | DS8000 | BRCD_ENCRYPTOR | ONESECURE | GENERIC | *userdevicegroup*} **-attribute** {*attributename*}]

Parameters

`-attribute`
Required. Specify an attribute for a device group. To determine the attributes of a device group, run the **`tklmDeviceGroupAttributeList`** command.

`device.AutoPendingAutoDiscovery`
Specifies whether to add a device that contacts IBM Security Key Lifecycle Manager to a list of pending devices that you can accept or reject before key serving occurs, or to add a device automatically to the drive table for immediate key service upon request. The attribute applies only to predefined base device families, not to user-defined device groups.

device.enableMachineAffinity
Specifies that device groups in the DS5000 device family are enabled to store the association of a device to an existing machine identifier in the IBM Security Key Lifecycle Manager database. Values are: `true` (enable) or `false` (disable). An instance of the property is stored for each device group.

drive.default.alias1
Specifies the system default certificate that a 3592 device uses if the device is not associated with another certificate.

drive.default.alias2
Specifies the system partner certificate that a 3592 device uses if the device is not associated with another certificate.

symmetricKeySet
Specifies a key group to be used for a device group.

shortName
This property specifies a short label that is usually a drive type such as LTO. This is used for any additional attributes that are required by an original equipment manufacturer.

longName
This property specifies an extended descriptive name of a drive type, such as `my division LTO`. For example, this information might include business information.

-name
Required. Specify a unique device group, such as LTO.

LTO Specifies the LTO device group.

3592 Specifies the 3592 device group.

DS5000
Specifies the DS5000 device group.

DS8000
Specifies the DS8000 device group.

BRCD_ENCRYPTOR
Specifies the BRCD_ENCRYPTOR device group that is in the LTO device family.

ONESECURE
Specifies the ONESECURE device group that is in the DS5000 device family.

GENERIC
Specifies a device family that uses the Key Management Interoperability Protocol to interact with IBM Security Key Lifecycle Manager. The GENERIC device group enables management of KMIP objects.

Do not use the command-line interface to add a device to the GENERIC device group, or to change a GENERIC device group attribute.

userdevicegroup
Specifies a user-defined group that is based on a supported device family.

Example

This Jython-formatted command deletes an attribute from a device group.

```
print AdminTask.tklmDeviceGroupAttributeDelete
('[-name myLTO -attribute "{symmetricKeySet}"]')
```

tklmDeviceGroupAttributeList

Use the **tklmDeviceGroupAttributeList** command to list all of the attributes of a device group such as LTO.

Note: The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager. Use the REST interfaces instead.

Purpose

Use this command to list all of the attributes of a device group such as LTO.

Permissions

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

Syntax

tklmDeviceGroupAttributeList -name {LTO | 3592 | DS5000 | DS8000 | BRCD_ENCRYPTOR | ONESECURE | GENERIC | *userdevicegroup*}

Parameters

-name

Required. Specify a unique device group, such as LTO.

LTO Specifies the LTO device group.

3592 Specifies the 3592 device group.

DS5000
Specifies the DS5000 device group.

DS8000
Specifies the DS8000 device group.

BRCD_ENCRYPTOR
Specifies the BRCD_ENCRYPTOR device group that is in the LTO device family.

ONESECURE
Specifies the ONESECURE device group that is in the DS5000 device family.

GENERIC
Specifies a device family that uses the Key Management Interoperability Protocol to interact with IBM Security Key Lifecycle Manager. The GENERIC device group enables management of KMIP objects.

Do not use the command-line interface to add a device to the GENERIC device group, or to change a GENERIC device group attribute.

userdevicegroup
Specifies a user-defined group that is based on a supported device family.

Example

This Jython-formatted command lists all of the attributes of a device group.

```
print AdminTask.tklmDeviceGroupAttributeList ('[-name 3592]')
```

tklmDeviceGroupAttributeUpdate

Use the **`tklmDeviceGroupAttributeUpdate`** command to update the attributes of a device group such as `myLTO`.

Note: The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager. Use the REST interfaces instead.

Purpose

Use this command to update the attributes of a device group such as `myLTO`. Use the **`tklmConfigUpdateEntry`** command to change an existing entry or to add an entry in the `SKLMConfig.properties` file.

Permissions

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

Syntax

tklmDeviceGroupAttributeUpdate [-name {LTO | 3592 | DS5000 | DS8000 | BRCD_ENCRYPTOR | ONESECURE | GENERIC | *userdevicegroup*} **-attributes** {*attributevaluepair*} {*attributevaluepair*}]

Parameters

`-attributes`
Specify one or more user-defined attribute-value pairs. Use the **`tklmDeviceGroupAttributeList`** command to view the current value.

`drive.default.alias1`
Specifies the system default certificate that a 3592 device uses if the device is not associated with another certificate.

`drive.default.alias2`
Specifies the system partner certificate that a 3592 device uses if the device is not associated with another certificate.

`enableKMIPDelete`
Enables or disables KMIP delete requests. The `klmAdminDeviceGroup` permission permits administration (create, view, delete) of a device group. Disabling this attribute when you create a device group prevents KMIP clients from deleting keys in the device group. The default is `disabled` (false). Use the **`tklmDeviceGroupAttributeUpdate`** command to modify this attribute.

`symmetricKeySet`
Specifies a key group to be used for a device group.

shortName
This property specifies a short label that is usually a drive type such as LTO. This is used for any additional attributes that are required by an original equipment manufacturer.

longName
This property specifies an extended descriptive name of a drive type, such as `my division LTO`. For example, this information might include business information.

-name
Required. Specify an existing device group, such as `myLTO`.

LTO Specifies the LTO device group.

3592 Specifies the 3592 device group.

DS5000
Specifies the DS5000 device group.

DS8000
Specifies the DS8000 device group.

BRCD_ENCRYPTOR
Specifies the BRCD_ENCRYPTOR device group that is in the LTO device family.

ONESECURE
Specifies the ONESECURE device group that is in the DS5000 device family.

GENERIC
Specifies a device family that uses the Key Management Interoperability Protocol to interact with IBM Security Key Lifecycle Manager. The GENERIC device group enables management of KMIP objects.

Do not use the command-line interface to add a device to the GENERIC device group, or to change a GENERIC device group attribute.

userdevicegroup
Specifies a user-defined group that is based on a supported device family.

Example

This Jython-formatted command updates an attribute for a 3592 device group.

```
print AdminTask.tklmDeviceGroupAttributeUpdate
('[-name 3592 -attributes "{longName 3592}"]')
```

This Jython-formatted command updates an attribute for an LTO device group.

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

tklmDeviceGroupBaseList

Use the **tklmDeviceGroupBaseList** command to list all of the device group families.

Note: The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager. Use the REST interfaces instead.

Purpose

Use this command to list all of the device group families.

Permissions

Your role must have a permission to the view action and a permission to at least one device group.

Syntax

tklmDeviceGroupBaseList

Parameters

There are no parameters.

Example

This Jython-formatted command lists all of device group families.

```
print AdminTask.tklmDeviceGroupBaseList ()
```

tklmDeviceGroupCreate

Use the **`tklmDeviceGroupCreate`** command to create a device group such as `myLTO`. The new device group is a child of a parent device family, such as LTO.

Note: The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager. Use the REST interfaces instead.

Purpose

Use this command to create a device group such as `myLTO`. The new device group is a child of a parent device family, such as LTO.

Permissions

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.

Syntax

tklmDeviceGroupCreate [-name *userdevicegroup* **-deviceFamily** {LTO | 3592 | DS5000 | GENERIC} **-attributes** {*attributevaluepair*} {*attributevaluepair*}]

Parameters

`-attributes`

Specify one or more user-defined attribute-value pairs. Use the **`tklmDeviceGroupAttributeList`** command to view the current value.

`device.enableMachineAffinity`

Specifies that device groups in the DS5000 device family are enabled to store the association of a device to an existing machine identifier in the IBM Security Key Lifecycle Manager database. Values are: `true` (enable) or `false` (disable). An instance of the property is stored for each device group.

`enableKMIPDelete`

Enables or disables KMIP delete requests. The `klmAdminDeviceGroup` permission permits administration (create, view, delete) of a device group. Disabling this attribute when you create a device group prevents KMIP clients from deleting keys in the device group. The default is `disabled` (false). Use the **`tklmDeviceGroupAttributeUpdate`** command to modify this attribute.

`shortName`

This property specifies a short label that is usually a drive type such as LTO. This is used for any additional attributes that are required by an original equipment manufacturer.

`longName`

This property specifies an extended descriptive name of a drive type, such as `my division LTO`. For example, this information might include business information.

`-deviceFamily`

Required. Specify an existing device family that IBM Security Key Lifecycle Manager provides.

LTO Specifies the LTO device family.

3592 Specifies the 3592 device family.

DS5000

Specifies the DS5000 device family.

GENERIC

Specifies a device family that uses the Key Management Interoperability Protocol to interact with IBM Security Key Lifecycle Manager. The GENERIC device group enables management of KMIP objects.

Do not use the command-line interface to add a device to the GENERIC device group, or to change a GENERIC device group attribute.

`-name`

Required. Specify a user-defined name for a device group. For example, specify `myDivisionLTO`.

Note: Do not specify a reserved value of 3592, DS8K, DS8000, LTO, DS5000, or GENERIC. Additionally, do not specify a reserved value of `SSLSERVER` or `SSLCLIENT`. The name must start with an alphabetic character, not a numeral, and can only contain alphanumeric characters and underscores. The name cannot consist of a single underscore and must not exceed a length of 16 characters.

Example

This Jython-formatted command creates a user-defined device group that has a customized attribute.

```
AdminTask.tklmDeviceGroupCreate ('[-name myLTO -deviceFamily LTO
 -attributes "{{shortName myLTO} {longName {my companyname LTO devices}} }"]')
```

The command returns a success message.

tklmDeviceGroupDelete

Use the **tklmDeviceGroupDelete** command to delete an empty customized device group such as myLTO. You cannot delete a device group if any devices, keys, or certificates are in that group. You also cannot delete a device family that IBM Security Key Lifecycle Manager provides.

Note: The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager. Use the REST interfaces instead.

Purpose

Use this command to delete an empty customized device group such as myLTO. You cannot delete a device group if any devices, keys, or certificates are in that group. You also cannot delete a device family that IBM Security Key Lifecycle Manager provides.

Permissions

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.

Syntax

tklmDeviceGroupDelete [**-name** *userdevicegroup*]

Parameters

-name

Required. Specify a user-defined name of an existing device group. You cannot delete a device family that a customized device group references.

Example

This Jython-formatted command deletes a user-defined device group.

```
print AdminTask.tklmDeviceGroupDelete ('[-name myLTO]')
```

The command returns true if the device group is deleted. Otherwise, the command returns false.

tklmDeviceGroupList

Use the **`tklmDeviceGroupList`** command to obtain a list of device groups within a device family such as LTO. If no values are specified, this command returns a list of all device groups.

Note: The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager. Use the REST interfaces instead.

Purpose

Use this command to obtain a list of device groups within a device family such as LTO. If no values are specified, this command returns a list of all device groups.

Permissions

Your role must have a permission to the view action and a permission to the appropriate device group, or your role must have a permission to the administrative actions.

Syntax

tklmDeviceGroupList -name *userdevicegroup* **-deviceFamily** {LTO | 3592 | DS5000 | DS8000 | GENERIC} **-v** {y | n}

Parameters

`-deviceFamily`
Optional. Specify a device family such as LTO.

LTO Specifies the LTO device family.

3592 Specifies the 3592 device family.

DS5000
Specifies the DS5000 device family.

GENERIC
Specifies a device family that uses the Key Management Interoperability Protocol to interact with IBM Security Key Lifecycle Manager. The GENERIC device group enables management of KMIP objects.

Do not use the command-line interface to add a device to the GENERIC device group, or to change a GENERIC device group attribute.

`-name`
Optional. Specify a user-defined name for a device group. For example, specify BRCD_ENCRYPTOR.

`-v [y | n]`
Verbose. The default is `n`, or no extra information. To list more information, specify `y` (for yes).

Example

This Jython-formatted command verbosely lists the device groups in the 3592 device family.

```
print AdminTask.tklmDeviceGroupList ('[-deviceFamily 3592 -v y]')
```

This Jython-formatted command shows that BRCD_ENCRYPTOR is a device group within the LTO device family.

```
print AdminTask.tklmDeviceGroupList ('[-name BRCD_ENCRYPTOR -v y]')
```

tklmGroupCreate

Use the **`tklmGroupCreate`** command to create a group to which you might add keys.

Note: The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager. Use the REST interfaces instead.

Purpose

Use this command to create a group to which you might add keys.

Permissions

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

Syntax

tklmGroupCreate -name *groupname* **-type** {*keygroup*} **-usage** {LTO | DS5000 | BRCD_ENCRYPTOR | ONESECURE | GENERIC | *userdevicegroup*}

Parameters

`-name`

Required. Specify a unique name for the group. The maximum value is 64 characters.

Note: The name of a DS5000 device key group is internally generated.

The characters in a name can include alphanumeric characters, and also the period, space, and underscore characters. However, unlike the graphical user interface, the command-line interface allows these special characters in a name:

```
~!@#$%^*()+|}{:?><`1234567890-=;/,
```

`-type`

Required. Specify the type of objects in the group. The value is not case-sensitive. You can specify the following values:

keygroup

The group contains keys.

`-usage`

Required. Specify a unique device group, such as LTO.

LTO Specifies the LTO device group.

DS5000

Specifies the DS5000 device group.

BRCD_ENCRYPTOR
Specifies the BRCD_ENCRYPTOR device group that is in the LTO device family.

ONESECURE
Specifies the ONESECURE device group that is in the DS5000 device family.

GENERIC
Specifies a device family that uses the Key Management Interoperability Protocol to interact with IBM Security Key Lifecycle Manager. The GENERIC device group enables management of KMIP objects.

Do not use the command-line interface to add a device to the GENERIC device group, or to change a GENERIC device group attribute.

userdevicegroup
Specifies a user-defined group that is based on a supported device family.

Examples

This Jython-formatted command creates a group to which you might add keys.

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

This Jython-formatted command creates a group that has a name that includes spaces.

```
print AdminTask.tklmGroupCreate
 ('[-name "my Key Group" -type keygroup -usage LTO]')
```

tklmGroupDelete

Use the **tklmGroupDelete** command to delete a key group. Deleting a populated key group *also deletes all the keys* in the key group.

Purpose

Use this command to delete a key group. Deleting a populated key group *also deletes all the keys* in the key group.

You cannot delete a key group that contains any keys that are assigned to a device. You must first remove the key assignments to those devices. You also cannot delete a key group that is scheduled for a future rollover. You must first delete the future rollover. You cannot delete a key group that is specified as the default symmetric key group by the value of the **symmetricKeySet** attribute for a device group. To delete this default group, you must first change the **symmetricKeySet** attribute value to a different group.

Permissions

Note: The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager. Use the REST interfaces instead.

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

Syntax

tklmGroupDelete -uuid *uuid_value*

Parameters

`-uuid`
Required. Specify a unique identifier for the group. For example, a key group uuid might be `GROUP-7d588437-e725-48bf-a836-00a47df64e78`.

Example

This Jython-formatted command deletes an existing key group *and also* the keys in the group.

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

tklmGroupEntryAdd

Use the **`tklmGroupEntryAdd`** command to add keys to an existing group.

Note: The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager. Use the REST interfaces instead.

Purpose

Use this command to add keys to an existing group.

Permissions

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

Syntax

tklmGroupEntryAdd -name *groupname* **-type** {keygroup} **-entry** {*attributevaluepair* }

Parameters

`-entry`
Required. Specify the entry to add to an existing group. You can specify the following attributes:

alias The unique name of a key. To add a key that you previously imported to a keystore, specify both the key and the keystore name.

keyStoreName
The unique name of the keystore.

type

key The entry object is a key.

uuid The Unique Universal Identifier of an entry. For example, the value might be `KEY-a3ce9230-bef9-42bd-86b7-6d208ec119cf`.

-name
Required. Specify a unique name of an existing group.

-type
Required. Specify the type of the specified group.

keygroup
The group contains keys.

Example

This Jython-formatted command adds a specific key to an existing key group.

```
print AdminTask.tklmGroupEntryAdd('[-name GROUP-myKeyGroup
 -type keygroup -entry "{type key}
  {alias aaa000000000000000000}
   {keyStoreName defaultKeyStore}"]')
```

tklmGroupEntryDelete

Use the **tklmGroupEntryDelete** command to delete objects from a group. For example, you might delete a key from membership in a key group. You can delete only one object at a time. This command does not delete the key metadata and key material.

Note: The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager. Use the REST interfaces instead.

Purpose

Use this command to delete objects from a group. For example, you might delete a key from membership in a key group. You can delete only one object at a time. This command does not delete the key metadata and key material.

Permissions

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

Syntax

tklmGroupEntryDelete -name *groupname* **-type** {keygroup} **-entry** {*attributevaluepair* }

Parameters

-entry
Required. Specify the entry to delete from an existing group: You can include the following attributes:

type

key The entry object is a key.

uuid Required. Specify the Unique Universal Identifier of an entry. For example, the value might be KEY-a3ce9230-bef9-42bd-86b7-6d208ec119cf.

-name
Required. Specify a unique name of an existing group.

-type
Required. Specify the type of the specified group.

keygroup
The group contains keys.

Example

This Jython-formatted command removes a specific key from membership in an existing group named GROUP-myKeyGroup.

```
print AdminTask.tklmGroupEntryDelete ('[-entry "{type key}
 {uuid KEY-a3ce9230-bef9-42bd-86b7-6d208ec119cf}"
   -name GROUP-myKeyGroup -type keygroup]')
```

tklmGroupList

Use the **tklmGroupList** command to list the objects in a group of keys, or the groups of a specific type.

Note: The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager. Use the REST interfaces instead.

Purpose

Use this command to list the objects in a group of keys, or the groups of a specific type.

Permissions

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

Syntax

tklmGroupList **-name** *groupname* **-type** {*keygroup*} **-usage** {LTO | DS5000 | BRCD_ENCRYPTOR | ONESECURE | GENERIC | *userdevicegroup*} **-v** {y | n}

Parameters

-name
Specify the name of the group.

-type
Required. Specify the type of objects in the group. The value is not case-sensitive.

keygroup
The group contains keys.

-usage
Specify a unique device group, such as LTO.

You can include the following values:

LTO Specifies the LTO device group.

DS5000
Specifies the DS5000 device group.

BRCD_ENCRYPTOR
: Specifies the BRCD_ENCRYPTOR device group that is in the LTO device family.

ONESECURE
: Specifies the ONESECURE device group that is in the DS5000 device family.

GENERIC
: Specifies a device family that uses the Key Management Interoperability Protocol to interact with IBM Security Key Lifecycle Manager. The GENERIC device group enables management of KMIP objects.

  Do not use the command-line interface to add a device to the GENERIC device group, or to change a GENERIC device group attribute.

userdevicegroup
: Specifies a user-defined group that is based on a supported device family.

-v [y | n]
: Verbose. The default is `n`, or no extra information. To list more information about a group, specify:

```
-v y
```

Example

This Jython-formatted command verbosely lists all keys in a group named myKeyGroup.

```
print AdminTask.tklmGroupList
 ('[-name GROUP-myKeyGroup -type keygroup -usage LTO -v y]')
```

tklmGroupUpdate

Use the **tklmGroupUpdate** command to update group metadata in the database to move all the keys in a key group from one device group to another within the same device group family.

Note: The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager. Use the REST interfaces instead.

Purpose

Use this command to update group metadata in the database to move all the keys in a key group from one device group to another within the same device group family.

Permissions

Your role must have a permission to the modify action, and permission to both the old and new device groups.

Syntax

tklmGroupUpdate -name *keygroupname* **-usage** {LTO | DS5000 | BRCD_ENCRYPTOR | ONESECURE | GENERIC | *userdevicegroup*} **-uuid** *keygroupID*

Parameters

-name
Specify the alias of the key group to update. You must specify a value for either **-name** or **-uuid**. If both are specified, the values must match.

-usage
Specify a unique device group, such as myNewLTO.

You can include the following values:

LTO Specifies the LTO device group.

DS5000
Specifies the DS5000 device group.

BRCD_ENCRYPTOR
Specifies the BRCD_ENCRYPTOR device group that is in the LTO device family.

ONESECURE
Specifies the ONESECURE device group that is in the DS5000 device family.

GENERIC
Specifies a device family that uses the Key Management Interoperability Protocol to interact with IBM Security Key Lifecycle Manager. The GENERIC device group enables management of KMIP objects.

Do not use the command-line interface to add a device to the GENERIC device group, or to change a GENERIC device group attribute.

userdevicegroup
Specifies a user-defined group that is based on a supported device family.

-uuid
Specify the Universal Unique Identifier of the key group. For example, GROUP-74386920-148c-47b2-a1e2-d19194b315cf might be the value. You must specify a value for either **-name** or **-uuid**. If both are specified, the values must match.

Example

This Jython-formatted command moves all the keys in the key group named myKeyGroup to the myNewLTO device group.

```
print AdminTask.tklmGroupUpdate('[-name myKeyGroup -usage myNewLTO]')
```

tklmKeyAttributeUpdate

Use the **tklmKeyAttributeUpdate** command to update key metadata that are Key Management Interoperability Protocol attributes in the database.

Note: The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager. Use the REST interfaces instead.

Purpose

Use this command to update key metadata that are Key Management Interoperability Protocol attributes in the database.

Permissions

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

Syntax

tklmKeyAttributeUpdate -uuid *keyuuid* **-operation** {add | update | delete} **-index** *indexofvalue* **-attrName** *attributename* **-attrValue** {*keyvaluepair* } {*keyvaluepair* }

Parameters

-attrName

Required. Specify the name that is used to identify or locate the attribute pair as an object.

Note: Do not use an asterisk (*) or question mark (?) as a character in a Key Management Interoperability Protocol attribute. These wildcard characters are reserved for future use.

applicationSpecificInformation

Specifies application namespace information as a Key Management Interoperability Protocol attribute.

contactInformation

Specifies contact information as a Key Management Interoperability Protocol attribute.

cryptoParams *cryptoparameter1*, *cryptoparameterN*

Specifies the cryptographic parameters that are used for cryptographic operations by using the object. This attribute is a Key Management Interoperability Protocol attribute.

customAttribute

Specifies a custom attribute in string format as a Key Management Interoperability Protocol attribute. Client-specific attributes must start with the characters "x-" (x hyphen) and server-specific attributes must start with "y-" (y hyphen).

link

Specifies the link from one managed cryptographic object to another, closely related target managed cryptographic object. This attribute is a Key Management Interoperability Protocol attribute.

name

Specifies the name that is used to identify or locate the object. This attribute is a Key Management Interoperability Protocol attribute.

objectGroup
Specifies one or more object group names of which this object might be part. This attribute is a Key Management Interoperability Protocol attribute.

processStartDate
Specifies the date on which a symmetric key object can be used for process purposes. You cannot change the value after the date occurs. If you specify a date earlier than the current date, the value is set to the current date. This attribute is a Key Management Interoperability Protocol attribute.

protectStopDate
Specifies the date on which an object cannot be used for process purposes. You cannot change the value after the date occurs. If you specify a date earlier than the current date, the value is set to the current date. This attribute is a Key Management Interoperability Protocol attribute.

usageLimits
Specifies either total bytes (BYTE) or total objects (OBJECT) as a Key Management Interoperability Protocol attribute. You cannot modify this value once this object is used. For example, **GetUsageAllocation** calls this object.

-attrValue
Specify one or more of these key value pairs:

applicationSpecificInformation *applicationIDstring*
Specifies application namespace information as the value of *applicationIDstring*.

NAMESPACE
Application namespace.

INFO
Application namespace information.

contactInformation *contactstring*
Specifies contact information as the value of *contactstring*. This attribute is a Key Management Interoperability Protocol attribute.

VALUE
Contact information.

cryptoParams *cryptoparameter1, cryptoparameterN*
Specifies the cryptographic parameters that are used for cryptographic operations by using the object. This attribute is a Key Management Interoperability Protocol attribute.

MODE
CBC, ECB, PCBC, CFB, OFB, CTR, CMAC, CCM, GCM, CBC_MAC, XTS, AES_KEY_WRAP_PADDING, NIST_KEY_WRAP, X9_102_AESKW, X9_102_TDKW, X9_102_AKW1, X 9_102_AKW2

PAD
NONE, OAEP, PKCS5, SSL3, ZEROS, ANSI_X9_23, ISO_10126, PKCS1_ V1_5, X9_31, PSS

HASH
MD2, MD4, MD5, SHA1, SHA224, SHA256, SHA384, SHA512

ROLE
BDK, CVK, DEK, MKAC, MKSMC, MKSMI, MKDAC, MKDN, MKCP, MKOTH,

`KEK, MAC1660 9, MAC97971, MAC97972, MAC97973, MAC97974, MAC97975, ZPK, PVKIBM, PVKPVV, PVKOTH`

customAttribute *customstring*
Specifies for the value of *customstring* a custom attribute in string format as a Key Management Interoperability Protocol attribute. Client-specific attributes must start with the characters "x-" (x hyphen) and server-specific attributes must start with "y-" (y hyphen).

NAME
Client or server attribute name.

VALUE
Value of the attribute name.

link *objectname*, *objectnametarget*
Specifies the link from one managed cryptographic object to another, closely related target managed cryptographic object. This attribute is a Key Management Interoperability Protocol attribute.

TYPE
`CERTIFICATE, PRIVATE_KEY, PUBLIC_KEY, DERIVATION_BASE_OBJECT, DERIVED_KEY, REPLACEMENT_OBJECT, REPLACED_OBJECT`

LINKED_OBJECT_ID
Specify the target uuid of the linked object.

name
Specifies the name that is used to identify or locate the object. This attribute is a Key Management Interoperability Protocol attribute.

TYPE
`TEXT, URI`

VALUE
Name, or `URI` identifying the object.

objectGroup *objectgroupname1*, *objectgroupnameN*
Specifies for *objectgroupname1, objectgroupnameN* the values of one or more object group names of which this object might be part. This attribute is a Key Management Interoperability Protocol attribute.

VALUE
Object group name.

processStartDate *yyyy-MM-dd*
Specifies the date in yyyy-MM-dd format on which a symmetric key object can be used for process purposes. You cannot change the value after the date occurs. If you specify a date earlier than the current date, the value is set to the current date. This attribute is a Key Management Interoperability Protocol attribute.

VALUE
Date in yyyy-MM-dd format.

protectStopDate *yyyy-MM-dd*
Specifies the date in yyyy-MM-dd format on which an object cannot be used for process purposes. You cannot change the value after the date occurs. If you specify a date earlier than the current date, the value is set to the current date. This attribute is a Key Management Interoperability Protocol attribute.

VALUE
Date in yyyy-MM-dd format.

usageLimits
Specifies either total bytes (BYTE) or total objects (OBJECT) as a Key Management Interoperability Protocol attribute. You cannot modify this value once this object is used. For example, **GetUsageAllocation** calls this object.

TYPE
OBJECT, BYTE

VALUE
Total number of objects or bytes.

index
Specify the index to update or delete an attribute value.

operation
Required. Specify one of these valid operations to run on an attribute value: add, update, delete

uuid
Required. Specify the Universal Unique Identifier of the key.

Example

This Jython-formatted command adds an attribute value of a key attribute.

```
print AdminTask.tklmKeyAttributeUpdate
 ('[-uuid KEY-d3ee4491-f96e-495d-bb37-fc03748924ba
  –operation add –attrName cryptoParams
   –attrValue  "{MODE CBC} {PAD NONE} {HASH SHA256} {ROLE BDK}"]')
```

This Jython-formatted command adds an attribute value of a key attribute.

```
print AdminTask.tklmKeyAttributeUpdate
 ('[-uuid KEY-d3ee4491-f96e-495d-bb37-fc03748924ba
  –operation add –attrName name
   -attrValue "{TYPE TEXT} {VALUE key name for xyz}"]')
```

This Jython-formatted command updates an attribute value of a key attribute.

```
print AdminTask.tklmKeyAttributeUpdate
 ('[-uuid KEY-d3ee4491-f96e-495d-bb37-fc03748924ba
  -operation update -index 0 -attrName name
   -attrValue "{TYPE TEXT} {VALUE updated key name for xyz}"]')
```

This Jython-formatted command deletes the value at the specified index.

```
print AdminTask.tklmKeyAttributeUpdate
 ('[-uuid KEY-d3ee4491-f96e-495d-bb37-fc03748924ba
  -operation delete -index 0 -attrName name]')
```

tklmKeyDelete

Use the **tklmKeyDelete** command to delete a key entry from the keystore. You cannot delete a key that is associated with a device or that is set as the default key for the device.

Note: The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager. Use the REST interfaces instead.

Purpose

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.

Use this command to delete a key entry from the keystore. You cannot delete a key that is associated with a device or that is set as the default key for the device.

The key and certificate states are changed to `destroyed` in the IBM Security Key Lifecycle Manager database, and the key material is deleted from the keystore.

Permissions

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

Syntax

tklmKeyDelete -alias *keyalias* **-keyStoreName** *keystorename*

Parameters

`-alias`
Required. Specify a unique name for the key.

`-keyStoreName`
Required. Specify the name of the keystore.

Example

This Jython-formatted command deletes a key entry from the keystore.

```
print AdminTask.tklmKeyDelete ('[-alias aaa000000000000000000
 -keyStoreName defaultKeyStore]')
```

tklmKeyExport

Use the **`tklmKeyExport`** command to export secret keys or public/private key pairs.

Note: The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager. Use the REST interfaces instead.

Purpose

Use this command 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.

Permissions

To export the key, you must have permission to the configure action, or get action plus the permission to the appropriate device group. When you export a secret key, you must also have permission to configure, create, or view action for the asymmetric key pair that is specified by the keyAlias parameter.

Syntax

tklmKeyExport -alias *keyalias* **-aliasRange** *prefixhexnumber1-hexnumberN* **-keyAlias** *keyalias* **-fileName** *pathandkeyfilename* **-keyStoreName** *keystorename* **-type** {secretkey | privatekey} **-password** *exportkeypasswordvalue*

Parameters

-alias
Required if a value is not specified for the aliasRange parameter. Specify the alias of the key to export. For a privatekey type, a value for -alias is required. For a secretkey type, you must specify a value for either **-alias** or **-aliasRange**.

-aliasRange
Required if a value is not specified for the alias parameter. When the value of alias is specified, the value of **-aliasRange** is ignored. Specify a three character prefix followed by a range of numbers in hexadecimal format (consisting of the sixteen characters 0-9, a-f) of secret keys to export. Allowed only for secret keys, not private keys. For example:

```
 -aliasRange ibm1-a
```

Alternatively, you might specify:

```
-aliasRange xyz01-fff
```

-fileName
Required. Specify the relative or full path, and the name of a file that IBM Security Key Lifecycle Manager creates to store the exported keys. If you do not specify a path name, the value of *SKLM_HOME* directory is used.

-keyAlias
Required if the exported key is a secret key. Specify the alias of the public key entry in the keystore that is used to encrypt the secret key, or keys, to the file. Only the holder of the corresponding private key can access the keys.

-keyStoreName
Required. Specify the name of the keystore from which the exported keys are exported.

-password
Required if the value of the **-type** attribute is `privatekey`. Specify a password to protect the PKCS#12 file to which the private key and certificate are exported.

You might want to retain the value of the password for later use with the **tklmKeyImport** command.

Note: If you migrate data from IBM Security Key Lifecycle Manager Version 1, any scripts or applications that you previously used to automate key export require modification to specify a password.

-type
Required. Specify whether the keys are secret or private. You can include the following values:

secretkey
Specifies a symmetric key.

privatekey
Specifies an asymmetric key in a key pair with a public key and a private key. If you select this value, a password is required. If you

export private keys to a PKCS#12 file, ensure that the file with the key is wrapped by using a FIPS-approved method before the file leaves the computer.

Example

For tape usage, this Jython-formatted command exports a range of secret keys into a file named mysecretkeys, in a path that is relative to the *SKLM_HOME* directory.

```
print AdminTask.tklmKeyExport ('[ -aliasRange abc1-ff
 -fileName mysecretkeys -keyStoreName defaultKeyStore
    -type secretkey -keyAlias mySecretKeyAlias]')
```

This Jython-formatted command specifies a password and exports a public/private key pair into a file named myprivatekeys, in a path that is relative to the *SKLM_HOME* directory.

```
print AdminTask.tklmKeyExport ('[ -alias myPrivateKeyAlias
 -fileName myprivatekeys -keyStoreName defaultKeyStore
    -type privatekey -password mypassword]')
```

tklmKeyGroupDefaultRolloverAdd

Use the **tklmKeyGroupDefaultRolloverAdd** command to add a default key group rollover to serve keys to a device group on a specific date. The rollover key group takes the place of the previous default key group.

Note: The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager. Use the REST interfaces instead.

Purpose

Use this command to add a default key group rollover to serve keys to a device group on a specific date. The rollover key group takes the place of the previous default key group.

Permissions

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

Syntax

tklmKeyGroupDefaultRolloverAdd -usage {LTO | BRCD_ENCRYPTOR | *userdevicegroup*} **-keyGroupName** *keygroupname* **-effectiveDate** *effectivedatevalue*

Parameters

-effectiveDate
Required. Specify the rollover date on which the key group becomes the default key group. The value is a current or future date in *yyyy-MM-dd* format. You cannot schedule two key groups for the same date as the default rollover.

-keyGroupName
Required. Specify the case-sensitive name of an existing key group.

-usage
Required. Specify the target application usage, such as an LTO device group. To find the available device groups, use the **tklmDeviceGroupList** command. You can specify the following values:

LTO Key group is used in secure communication with LTO tape drives.

BRCD_ENCRYPTOR
Specifies the BRCD_ENCRYPTOR device group that is in the LTO device family.

userdevicegroup
Specifies a new, user-defined instance of the LTO device family.

The value cannot exceed 16 characters in length. For example: `myLTO`.

Example

This Jython-formatted command specifies an existing, default key group.

```
print AdminTask.tklmKeyGroupDefaultRolloverAdd
('[-usage LTO -keyGroupName myLTOkeygroup
     -effectiveDate 2010-04-30]')
```

tklmKeyGroupDefaultRolloverDelete

Use the **tklmKeyGroupDefaultRolloverDelete** command to remove an existing key group rollover from the rollover list.

Note: The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager. Use the REST interfaces instead.

Purpose

Use this command to remove an existing key group rollover from the rollover list.

Permissions

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

Syntax

tklmKeyGroupDefaultRolloverDelete -uuid *keygrouprolloveruuid*

Parameters

-uuid
Required. Specify the Universal Unique Identifier of an existing key group rollover. Use the **tklmKeyGroupDefaultRolloverList()** command to list attributes, including the value of the **-uuid** parameter. For example: 201

Example

This Jython-formatted command removes a key group rollover from the rollover list.

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

tklmKeyGroupDefaultRolloverList

Use the **`tklmKeyGroupDefaultRolloverList`** command to list key group rollovers in a rollover list.

Note: The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager. Use the REST interfaces instead.

Purpose

Use this command to list key group rollovers in a rollover list.

Permissions

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

Syntax

tklmKeyGroupDefaultRolloverList -usage {LTO | BRCD_ENCRYPTOR | *userdevicegroup*} **-uuid** *keygrouprolloveruuid* **-name** *keygroupname* **-v** {y | n}

Parameters

`-name`
Optional. Specify the case-sensitive name of a key group. For example: `myLTO` key group

`-usage`
Required. Specify a device group such as an LTO device group. You can include the following values:

LTO Key is used in secure communication with LTO tape drives.

BRCD_ENCRYPTOR
Specifies the BRCD_ENCRYPTOR device group that is in the LTO device family.

userdevicegroup
Specifies a new, user-defined instance of the LTO device family.

The value cannot exceed 16 characters in length. For example: `myLTO`.

`-uuid`
Optional. Specify the Universal Unique Identifier of an existing key group rollover. For example: 201

`-v [y | n]`
Optional. Verbose. The default is `n`, or no extra information. To list more information about a key group, specify:

```
-v y
```

Example

This Jython-formatted command lists the key group rollovers in a rollover list.

```
print AdminTask.tklmKeyGroupDefaultRolloverList
('[-usage LTO]')
```

tklmKeyImport

Use the **tklmKeyImport** command 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 with a public key and a private key. The private key file is in PKCS#12 format.

Note: The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager. Use the REST interfaces instead.

Purpose

Use this command 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 with a public key and a private key. The private key file is in PKCS#12 format.

To import secret keys, the import file can contain multiple keys. Each key contains the required alias value for that key. The import file must be generated by a previous **tklmKeyExport** command.

Permissions

To import the key, you must have permission to the configure action, or, create action and the permission to the appropriate device group. When you import a secret key, you must also have permission to configure, create, or view action for the asymmetric key pair that is specified by the keyAlias parameter.

Syntax

An asterisk (*) indicates a deprecated value. If you enter the deprecated value, do not include the asterisk.

tklmKeyImport -alias *keyalias* **-newAlias** *newkeyalias* **-keyAlias** *keyalias* **-fileName** *pathandkeyfilename* **-keyStoreName** *keystorename* **-usage** {LTO | 3592 | DS5000 | DS8000 | BRCD_ENCRYPTOR | ONESECURE | GENERIC | *userdevicegroup* | SSLSERVER | SSL Server* | SSLCLIENT } **-type** {secretkey | privatekey} **-password** *passwordvalue*

Parameters

-alias
Required if the value of the **-type** attribute is secretkey and you want to rename the key during import by using the **-newAlias** parameter. Specify the value of alias if you want to import only this secret key from a secret key file that has other secret keys that you do not want to import.

A value for **-alias** is not required to import a private key because there is only one private key in the file. If you specify this value when you import a private key, the value is ignored.

-fileName
Required. Specify the path and file name of the file from which keys are imported.

-keyAlias
Required if the value of the **-type** attribute is secretkey. Specify the alias of the

private key entry in the keystore that is used to decrypt the secret key, or keys, from the file. Use the same alias value to import and export a secret key, or keys.

-keyStoreName
Required. Specify the name of the keystore into which the imported key is imported.

-newAlias
Specify a new value for the key alias.

-password
Required if the value of the **-type** attribute is `privatekey`. This password was previously specified by using the **tklmKeyExport** command. If you export private keys to a `PKCS#12` file, ensure that the file with the key is wrapped by using a FIPS-approved method before the file leaves the computer.

-type
Required. Specify whether the keys are secret or private. You can include the following values:

secretkey
Specifies a symmetric key.

If you select this value, specify for the **-usage** attribute a value for a device group family that administers keys.

privatekey
Specifies an asymmetric key in a key pair with a public key and a private key.

If you select this value, specify for the **-usage** attribute a value for a device group that administers certificates, or specify one of these values:

- SSLCLIENT
- SSLSERVER

-usage
Required. Specify the target application usage, such as LTO device group. You can include the following values:

LTO Specifies the LTO device group.

3592 Specifies the 3592 device group.

DS5000
Specifies the DS5000 device group.

DS8000
Specifies the DS8000 device group.

BRCD_ENCRYPTOR
Specifies the BRCD_ENCRYPTOR device group that is in the LTO device family.

ONESECURE
Specifies the ONESECURE device group that is in the DS5000 device family.

GENERIC
Specifies a device family that uses the Key Management

Interoperability Protocol to interact with IBM Security Key Lifecycle Manager. The GENERIC device group enables management of KMIP objects.

Do not use the command-line interface to add a device to the GENERIC device group, or to change a GENERIC device group attribute.

SSLCLIENT
: Client-side certificate that is used in secure communication by using Secure Socket Layer protocol to authenticate the client device.

SSLSERVER
: Server-side certificate that is used in secure communication by using Secure Socket Layer protocol.

userdevicegroup
: Specifies a user-defined group that is based on a supported device family.

Example

For tape usage, this Jython-formatted command imports a symmetric key named mysecretkey from a file named mykey.p12 into a keystore named myKeystore for use with LTO tape drives.

```
print AdminTask.tklmKeyImport ('[ -alias mysecretkey -type secretkey
 -keyAlias mySecretKey -newAlias myNewSecretKey
  -fileName c:\\myimportpath\\mykey.p12 -keyStoreName defaultKeyStore
   -usage LTO]')
```

This Jython-formatted command imports an asymmetric key in a key pair with a public key and a private key. A password is required.

```
print AdminTask.tklmKeyImport ('[-alias myprivatekey -type privatekey
 -keyAlias myPrivateKeyAlias -fileName c:\\myimportpath\\myprivatekey.p12
  -keyStoreName defaultKeyStore -usage SSLSERVER -password mypassword]')
```

tklmKeyList

Use the **tklmKeyList** command to list a key or keys in the keystore, which is based on specified criteria such as an active state.

Note: The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager. Use the REST interfaces instead.

Purpose

Use this command to list a key or keys in the keystore, which is based on specified criteria such as an active state.

Permissions

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

Syntax

tklmKeyList -uuid *universalKeyID* **-alias** *keyalias* **-keyStoreName** *keystorename* **-usage** {LTO | 3592 | DS5000 | DS8000 | BRCD_ENCRYPTOR | ONESECURE | GENERIC | *userdevicegroup* | SSLSERVER | SSLCLIENT } **-attributes** [state *value*] **-v** {y | n}

Note: The KMIP interface for IBM Security Key Lifecycle Manager Version 2 does not support private and public key objects.

IBM Security Key Lifecycle Manager creates certificate objects with public and private key objects internally, but does not set some KMIP required and optional attributes on these objects. Running the **tklmKeyList** command in verbose mode lists many of the KMIP attributes as NULL. Similarly, symmetric key objects that are created through the non-KMIP interface of IBM Security Key Lifecycle Manager list many of the KMIP attributes as NULL. The null values do not affect the IBM Security Key Lifecycle Manager function.

Parameters

There are no required parameters.

`-alias`
Specify a unique name for the key.

`-attributes`
Specify the current state of the key. The following values are supported:

`pending`
A certificate request entry is pending the return of a certificate that is approved and certified by a certificate authority.

`pre-active`
Object exists but is not yet usable for any cryptographic purpose, such as migrated certificates with a future use time stamp.

`active`
Object is in operational use for protecting and processing data that might use **`Process Start Date`** and **`Protect Stop Date`** attributes. For example, protecting includes encryption and signature issue. Processing includes decryption and signature verification.

`compromised`
The security of the object is suspect for some reason. A compromised object never returns to an `uncompromised` state, and cannot be used to protect data. Use the object only to process cryptographically protected information in a client that is trusted to handle compromised cryptographic objects.

IBM Security Key Lifecycle Manager retains the state of the object immediately before it was compromised. To process data that was previously protected, the compromised object might continue to be used.

`deactivated`
Object is not to be used to apply cryptographic protection such as encryption or signing. However, if extraordinary circumstances occur, the object can be used with special permission to process cryptographically protected information. For example, processing includes decryption or verification.

destroyed
Object is no longer usable for any purpose. This status causes the object to be removed from the product.

destroyed-compromised
Object is no longer usable for any purpose. This status causes the object to be removed from the product.

-keyStoreName
Specify the name of the keystore.

-uuid
Specify the Universal Unique Identifier of the key. For example, `KEY-a3ce9230-bef9-42bd-86b7-6d208ec119cf`.

-usage
Specify a unique device group, such as LTO.

You can include the following values:

LTO Specifies the LTO device group.

3592 Specifies the 3592 device group.

DS5000
Specifies the DS5000 device group.

DS8000
Specifies the DS8000 device group.

ONESECURE
Specifies the ONESECURE device group that is in the DS5000 device family.

GENERIC
Specifies a device family that uses the Key Management Interoperability Protocol to interact with IBM Security Key Lifecycle Manager. The GENERIC device group enables management of KMIP objects.

Do not use the command-line interface to add a device to the GENERIC device group, or to change a GENERIC device group attribute.

SSLCLIENT
Client-side certificate that is used in secure communication by using Secure Socket Layer protocol to authenticate the client device.

SSLSERVER
Server-side certificate that is used in secure communication by using Secure Socket Layer protocol.

userdevicegroup
Specifies a user-defined group that is based on a supported device family.

-v [y | n]
Verbose. The default is `n`, or no extra information. To list more information about a key, specify `y` (for yes):

```
-v y
```

Example

This Jython-formatted command verbosely lists all the keys that are in active state.

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

tklmKeyUpdate

Use the **tklmKeyUpdate** command to update key metadata in the database. For example, you might move an individual key in one key group to another key group.

Note: The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager. Use the REST interfaces instead.

Purpose

Use this command to update key metadata in the database. For example, you might move an individual key in one key group to another key group.

Permissions

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

Syntax

tklmKeyUpdate -uuid *universalKeyID* **-usage** {LTO | 3592 | DS5000 | DS8000 | BRCD_ENCRYPTOR | ONESECURE | GENERIC | *userdevicegroup*} **-attributes** {*attributevaluepair*}{*attributevaluepair*}

Parameters

-attributes

Specify one or more of these attribute-value pairs:

compromised

Specifies whether the use of a key is compromised. The only value is y (compromised). You cannot change a compromised key or certificate to an uncompromised state.

groupName

Specifies the name of a valid key group. You cannot move the last key in a default key group to a different group.

However, you can change the key group name to a key group used by a different device group in the same device family, if the key group and any of its keys are not the device group default, or attached to a device. For example, you can change such a group from the myLTO device group to yourLTO device group in the LTO device family.

In the DS5000 device family, a key group is generated for each DS5000 device when the device is created. You cannot create a DS5000 device with a key group attribute. However, you can create a new key group and specify the group name of a DS5000 device with the new key group.

information *informationstring*

Specifies more information about the use of an object.

-uuid

Required. Specify the Universal Unique Identifier of the individual key that you want to move.

-usage
Specify a unique device group, such as LTO.

You can include the following values:

LTO Specifies the LTO device group.

3592 Specifies the 3592 device group.

DS5000
Specifies the DS5000 device group.

DS8000
Specifies the DS8000 device group.

BRCD_ENCRYPTOR
Specifies the BRCD_ENCRYPTOR device group that is in the LTO device family.

ONESECURE
Specifies the ONESECURE device group that is in the DS5000 device family.

GENERIC
Specifies a device family that uses the Key Management Interoperability Protocol to interact with IBM Security Key Lifecycle Manager. The GENERIC device group enables management of KMIP objects.

Do not use the command-line interface to add a device to the GENERIC device group, or to change a GENERIC device group attribute.

userdevicegroup
Specifies a user-defined group that is based on a supported device family.

Example

This Jython-formatted command updates the metadata for a key to indicate that the status of the key is compromised and describes the date of compromise.

```
print AdminTask.tklmKeyUpdate ('[-uuid KEY-a3ce9230-bef9-42bd-86b7-6d208ec119cf
 -usage LTO -attributes "{compromised y} {information compromised_052208}"]')
```

tklmKeyServerStatus

Use the **tklmKeyServerStatus** command to return the status of the key server, an internal component that the IBM Security Key Lifecycle Manager server contains. You can use this command to confirm that IBM Security Key Lifecycle Manager server is properly configured to serve keys to particular device groups.

Note: The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager. Use the REST interfaces instead.

Purpose

Use this command to return the status of the key server, an internal component that the IBM Security Key Lifecycle Manager server contains.

You can use this command to confirm that IBM Security Key Lifecycle Manager server is properly configured to serve keys to particular device groups. IBM Software Support might request that you run this command for diagnostic purposes. For example, the IBM Security Key Lifecycle Manager server might remain running when the key server component is not running.

Permissions

There is no role or permission restriction on this action.

Syntax

tklmKeyServerStatus

Parameters

There are no parameters.

Example

This Jython-formatted command returns the current state of the key server.

```
print AdminTask.tklmKeyServerStatus ()
```

tklmKeyStoreAdd

Use the **`tklmKeyStoreAdd`** command to add a file-based keystore.

Note: This IBM Security Key Lifecycle Manager command-line interface command is deprecated.

Purpose

Use this command to add a keystore. If the keystore does not exist, and the keystore type is file-based, IBM Security Key Lifecycle Manager creates the keystore. If one keystore exists, you cannot create an extra keystore. Do not add a keystore file with a name that is identical to the file name of a deleted keystore.

Permissions

Your role must have a permission to the configure action.

Syntax

tklmKeyStoreAdd -storeName *keystorename* **-storeFileName** *filelocationandname* **-storeType** *keystoretype* **-storePassword** *keystorepassword*

Parameters

`-storeName`
Required. Specify the unique name of the keystore as a descriptive alias. For example, type `newKeystore`. IBM Security Key Lifecycle Manager uses this name in the IBM Security Key Lifecycle Manager database to identify the keystore.

`-storeFileName`
Required. How IBM Security Key Lifecycle Manager uses this value depends on the keystore type:

- File-based

  Specifies the directory path and file name of the keystore file.

-storeType
Specify the type of the keystore, such as JCEKS (default).

-storePassword
Specify the password of the keystore file. For example, password. The password in single-byte characters must a minimum of 6 and not greater than 175 characters in length.

Example

This Jython-formatted command adds a keystore named newKeyStore.jceks, which IBM Security Key Lifecycle Manager identifies in the database by the name newKeyStore.

```
print AdminTask.tklmKeyStoreAdd ('[-storeName newKeyStore
  -storeFileName SKLM_HOME/keystore/
newKeyStore
  -storeType jceks -storePassword password]')
```

tklmKeyStoreDelete

Use the **tklmKeyStoreDelete** command to delete the alias for a file-based keystore from the IBM Security Key Lifecycle Manager database. For example, only during a test phase, you might delete a keystore because you want to use a different keystore. **Do not delete a keystore that is used in production.**

Note: This IBM Security Key Lifecycle Manager command-line interface command is deprecated.

Purpose

Use this command to delete the alias for a file-based keystore from the IBM Security Key Lifecycle Manager database. For example, only during a test phase, you might delete a keystore because you want to use a different keystore. **Do not delete a keystore that is used in production.**

Note:

- IBM Security Key Lifecycle Manager deletes both the alias for the keystore and metadata for keys and certificates from the IBM Security Key Lifecycle Manager database. The command does not delete the physical keystore or key material.
- When you delete a keystore, IBM Security Key Lifecycle Manager does *not* delete devices that have associated certificates, keys, or key groups in the deleted keystore. After you create a new keystore, you must associate new certificates, keys, or key groups with previously existing devices.

  Keys and key groups are automatically generated the first time that you create DS5000 devices by using the graphical user interface. If you later delete the keystore, you cannot repeat the graphical user interface step to create DS5000 devices that exist. You must instead use the command-line interface to associate keys and key groups to each existing DS5000 device, a time-consuming process.
- After you delete a keystore, do not create a new keystore with a name that is identical to a deleted keystore.
- Restoring a deleted keystore requires a current backup of the deleted IBM Security Key Lifecycle Manager keystore. After you delete a keystore, restart the WebSphere Application Server.

Permissions

Your role must have a permission to the configure action.

Syntax

tklmKeyStoreDelete -storeName *keystorename* **-confirm** {y| n}

Parameters

-storeName
Required. Specify the unique name of the keystore as a descriptive alias. For example, type `myKeystore`. IBM Security Key Lifecycle Manager uses this name as an alias in the database to identify the keystore.

-confirm [y | n]
Required. Confirms keystore deletion. The default is `n`.

Example

This Jython-formatted command deletes the `myKeystore` alias and metadata from the IBM Security Key Lifecycle Manager database for a physical keystore named `myKeystore.jceks`.

```
print AdminTask.tklmKeyStoreDelete ('[-storeName myKeystore
  -confirm y]')
```

This Jython-formatted command deletes a keystore that has a name with spaces.

```
print AdminTask.tklmKeyStoreDelete
 ('[-storeName "my Keystore" -confirm y]')
```

tklmKeyStoreEntryMetaDataCreate

Use the **`tklmKeyStoreEntryMetaDataCreate`** command to create and store metadata about an existing asymmetric key in an existing keystore.

Note: This IBM Security Key Lifecycle Manager command-line interface command is deprecated.

Purpose

Use this command to create and store metadata about an existing asymmetric key in an existing keystore.

Permissions

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

Syntax

An asterisk (*) indicates a deprecated value. If you enter the deprecated value, do not include the asterisk.

tklmKeyStoreEntryMetaDataCreate -alias *keyalias* **-type** *privatekey* **-usage** {3592 | DS8000 | DS8K* | GENERIC | *userdevicegroup* | SSLSERVER | SSLCLIENT | IKEV2SERVER | IKEV2CLIENT} **-keyStoreName** *keystorename*

Parameters

-alias
Required. Specify a unique name for the key in the keystore.

-type
Required. Specify `privatekey` as the key type.

-usage
Optional. Specify the target usage, such as 3592 (the default, if no value is specified). You can specify the following values:

3592 Certificate is used in secure communication for 3592 tape drives.

DS8000
Certificate is used in secure communication for DS8000 Turbo drives.

GENERIC
Certificate is used in secure communication for a device that uses Key Management Interoperability Protocol.

Do not use the command-line interface to add a device to the GENERIC device group, or to change a GENERIC device group attribute.

SSLCLIENT
Client-side certificate that is used in secure communication by using Secure Socket Layer protocol to authenticate the client device.

SSLSERVER
Server-side certificate that is used in secure communication by using Secure Socket Layer protocol.

userdevicegroup
Specifies a user-defined group that is based on a supported device family.

-keyStoreName
Required. Specify the name of an existing keystore.

Example

This Jython-formatted command creates a metadata entry in the keystore for an existing asymmetric key.

```
print AdminTask.tklmKeyStoreEntryMetaDataCreate
 ('[-alias abc -type privatekey -keyStoreName defaultKeyStore -usage DS8000]')
```

tklmKeyStoreList

Use the **tklmKeyStoreList** command to list the IBM Security Key Lifecycle Manager keystore.

Note: This IBM Security Key Lifecycle Manager command-line interface command is deprecated.

Purpose

Use this command to list the IBM Security Key Lifecycle Manager keystores.

Permissions

Your role must have a permission to the configure action.

Syntax

tklmKeyStoreList -storeName *keystorename* **-storeUuid** *keystoreuuid* **-v** {y| n}

Parameters

There are no required parameters. If you do not specify a value for either a keystore name or the Universal Unique Identifier, the command lists all keystores.

-storeName
: Specify the unique name of the keystore. IBM Security Key Lifecycle Manager uses this name in the database to identify the keystore.

-storeUuid
: Specify the Universal Unique Identifier of the keystore. If you specify both values, this value is used first to query the keystore table.

-v [y | n]
: Verbose. The default is `n`, or no extra information. To list more information about a keystore, specify:

```
-v y
```

Example

This Jython-formatted command lists information available for a keystore named defaultKeyStore.

```
print AdminTask.tklmKeyStoreList ('[-storeName defaultKeyStore -v y]')
```

This command lists the existing keystore.

```
print AdminTask.tklmKeyStoreList()
```

tklmKeyStoreUpdate

Use the **tklmKeyStoreUpdate** command to update the name and password of an existing file-based keystore. When you change the password for the keystore, you also change the password for all the keys in the keystore.

Note: This IBM Security Key Lifecycle Manager command-line interface command is deprecated.

Purpose

Use this command to update the name and password of an existing file-based keystore. When you change the password for the keystore, you also change the password for all the keys in the keystore. Your role must have a permission to the configure action.

Permissions

Your role must have a permission to the configure action.

Syntax

tklmKeyStoreUpdate -storeName *keystorename* **-attributes** {*attributevaluepair* } {*attributevaluepair* } **-storePassword** *keystorepassword*

Parameters

-attributes
Specify one or more attribute-value pairs:

-storeName
Specify the *new* name of the existing keystore.

Note: In a running production environment, *do not modify* the keystore name. If you must modify the keystore name before production, ensure that you have a complete, current backup of your keystore.

-storePassword
Specify the *new* password of the keystore file. When you change the password for the keystore, you also change the password for all the keys in the keystore.

-storeName
Required. Specify the *current* name of the existing keystore. IBM Security Key Lifecycle Manager uses this name in the database to identify the keystore. This value is determined when you first install and configure IBM Security Key Lifecycle Manager.

-storePassword
Required. Specify the *current* password of the keystore file. For example, `password`. The password must be at least 6 characters in length.

Example

This Jython-formatted command changes the keystore password.

```
print AdminTask.tklmKeyStoreUpdate ('[-storeName defaultKeyStore
   -storePassword currentstorepwd -attributes "{storePassword newstorepwd}"]')
```

tklmKMIPTemplateDelete

Use the **tklmKMIPTemplateDelete** command to delete templates. For example, you might delete a template that a client registered.

Note: The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager. Use the REST interfaces instead.

Purpose

Use this command to delete templates.

Permissions

You must have the `klmSecurityOfficer` role.

Syntax

tklmKMIPTemplateDelete -uuid *templateuuid*

Parameters

-uuid
Required. Specify the unique universal identifier of an existing template. For example: `K_TEMPLATE-d3ee4491-f96e-495d-bb37-fc03748924ba`

Example

This Jython-formatted command deletes a template.

```
print AdminTask.tklmKMIPTemplateDelete
 ('[-uuid K_TEMPLATE-d3ee4491-f96e-495d-bb37-fc03748924ba]')
```

tklmKMIPTemplateList

Use the **`tklmKMIPTemplateList`** command to list KMIP templates that IBM Security Key Lifecycle Manager provides. For example, you might list all templates.

Note: The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager. Use the REST interfaces instead.

Purpose

Use this command to list KMIP templates that IBM Security Key Lifecycle Manager provides. For example, you might list all templates that are created or registered by the client.

Permissions

You must have the `klmSecurityOfficer` role.

Syntax

tklmKMIPTemplateList -uuid *templateuuid* **-v** {y| n}

Parameters

`-uuid`
Optional. Specify the unique universal identifier of an existing template. For example: `K_TEMPLATE-d3ee4491-f96e-495d-bb37-fc03748924ba`

`-v [y | n]`
Optional. Verbose. The default is `n`, or no additional information other than the **-uuid** and **-name** attribute. To list more information about a template, specify:

```
-v y
```

Example

This Jython-formatted command verbosely lists a template.

```
print AdminTask.tklmKMIPTemplateList
 ('[-uuid K_TEMPLATE-d3ee4491-f96e-495d-bb37-fc03748924ba -v y]')
```

The output is as shown in the following example:

```
Unique Identifier = K_TEMPLATE-d3ee4491-f96e-495d-bb37-fc03748924ba
Cryptographic Algorithm = TRIPLE_DES
Cryptographic Length = 168
Operation Policy Name = default policy for template
Cryptographic Usage Mask = ENCRYPT
Usage Limits = [[TYPE OBJECT] [TOTAL 1000]]
Activation Date = 2009-12-01
Process Start Date = 2009-12-05
Protect Stop Date = 2010-11-05
Deactivation Date = 2010-12-01
Contact Information = 1540 Scenic Ave, Costa Mesa, CA 92626
Name = [[INDEX 1] [TYPE TEXT] [VALUE template for triple des symmetric key]],
[[INDEX 2] [TYPE URI] [VALUE template for xyz]]
Cryptographic Parameters = [[INDEX 1] [MODE CBC]
```

```
[PAD ISO_10126] [HASH SHA256] [ROLE ZMK]]
Object Group = [[INDEX 1] [VALUE object group 1]], [[INDEX 2]
[VALUE object group2]]
Application Specific Information =
[[INDEX 1] [NAMESPACE HYPERLINK "https://www.ibm.com/tklm%5D
%5BINFO"https://www.ibm.com/tklm] [INFO used for tklm]],[[ INDEX 2]
[NAMESPACE https://www.ibm.com/tklm1] [INFO used for tklm1]]
Custom Attribute = [[NAME x-clientattr1] [[INDEX 0] [TYPE TEXT] [VALUE value1]]],
[[NAME x-clientattr2] [[[INDEX 0] [TYPE TEXT] [VALUE value2]]],
[[NAME y-serverattr1] [[[INDEX 0] [TYPE TEXT] [VALUE value3]], [[INDEX 1] [VALUE value4]]]]
Initial Date = 2009-12-01
Last Change Date = 2009-12-01
Destroy Date =
Archive Date =
Owner = base64 encoded SHA-256 hash public key of client certificate
```

tklmMachineDeviceAdd

Use the **tklmMachineDeviceAdd** command to add the association of a device to an existing machine identifier in the IBM Security Key Lifecycle Manager database.

Note: The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager. Use the REST interfaces instead.

Purpose

Use this command to add the association of a device to an existing machine identifier in the IBM Security Key Lifecycle Manager database.

Permissions

Your role must have a permission to the create action and a permission to the DS5000 device group.

Syntax

tklmMachineDeviceAdd -deviceUUID *deviceuuid* **-machineID** *machineidentitystring* | **-machineText** *machineidentitytext*

Parameters

-deviceUUID
Specify a value for a unique universal identifier for the device, such as DEVICE-7d588437-e725-48bf-a836-00a47df64e78. Use the **tklmDeviceList** command to locate the device uuid.

-machineID
Required if the value of **-machineText** is not specified. Specify a unique machine ID with a minimum length of 1 and a maximum length of 48 characters. For example: 3042383030303437000000000000. Use the **tklmMachineIdentityList** command to locate machine identities.

-machineText
Required if the value of **-machineID** is not specified. Specify a unique, user-supplied descriptive label greater than zero bytes and not more than 96 bytes for a machine, such as myEncryptedDS5000.

Example

This Jython-formatted command adds the association of a device to a known, existing machine ID in the IBM Security Key Lifecycle Manager database.

```
print AdminTask.tklmMachineDeviceAdd
('[-deviceUUID DEVICE-7d588437-e725-48bf-a836-00a47df64e78
-machineID 30423830303034370000000000000]')
```

tklmMachineDeviceDelete

Use the **tklmMachineDeviceDelete** command to remove the association of a device from an existing machine identifier in the IBM Security Key Lifecycle Manager database.

Note: The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager. Use the REST interfaces instead.

Purpose

Use this command to remove the association of a device from an existing machine identifier in the IBM Security Key Lifecycle Manager database.

Permissions

Your role must have a permission to the delete action and a permission to the DS5000 device group.

Syntax

tklmMachineDeviceDelete -deviceUUID *deviceuuid* **-machineID** *machineidentitystring* **-machineText** *machineidentitytext*

Parameters

-deviceUUID
Specify a value for a unique universal identifier for the device, such as DEVICE-7d588437-e725-48bf-a836-00a47df64e78. Use the **tklmDeviceList** command to locate the device uuid.

-machineID
Required if the value of **-machineText** is not specified. Specify a unique machine ID with a minimum length of 1 and a maximum length of 48 characters. For example: 30423830303034370000000000000. Use the **tklmMachineIdentityList** command to locate machine identities.

-machineText
Required if the value of **-machineID** or **-deviceUUID** is not specified. Specify a unique, user-supplied descriptive label greater than zero bytes and not more than 96 bytes for a machine, such as myEncryptedDS5000.

Example

This Jython-formatted command removes the association of a device from a known, existing machine ID in the IBM Security Key Lifecycle Manager database.

```
print AdminTask.tklmMachineDeviceDelete
('[-deviceUUID  DEVICE-7d588437-e725-48bf-a836-00a47df64e78
-machineID 30423830303034370000000000000]')
```

tklmMachineDeviceList

Use the **tklmMachineDeviceList** command to list all the devices that are associated with a specific machine ID.

Note: The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager. Use the REST interfaces instead.

Purpose

Use this command to list all the devices that are associated with a specific machine ID.

Permissions

Your role must have a permission to the view action and a permission to the DS5000 device group.

Syntax

tklmMachineDeviceList -machineID *machineidentitystring* **-machineText** *machineidentitytext*

Parameters

`-machineID`
Required if the value of **`-machineText`** is not specified. Specify a unique machine ID with a minimum length of 1 and a maximum length of 48 characters. For example: `3042383030303437000000000000`. Use the **`tklmMachineIdentityList`** command to locate machine identities.

`-machineText`
Required if the value of **`-machineID`** is not specified. Specify a unique, user-supplied descriptive label greater than zero bytes and not more than 96 bytes for a machine, such as `myEncryptedDS5000`.

Example

This Jython-formatted command lists all the devices that are associated with a specific machine ID.

```
print AdminTask.tklmMachineDeviceList
('[-machineID 3042383030303437000000000000]')
```

tklmMachineIdentityAdd

Use the **`tklmMachineIdentityAdd`** command to create a machine identity in the IBM Security Key Lifecycle Manager database.

Note: The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager. Use the REST interfaces instead.

Purpose

Use this command to create a machine identity in the IBM Security Key Lifecycle Manager database.

Permissions

Your role must have a permission to the create action and a permission to the DS5000 device group to which the machine identity is attached.

Syntax

tklmMachineIdentityAdd -machineID *machineidentitystring* | **-machineText** *machineidentitytext*

Parameters

-machineID
Required if the value of **-machineText** is not specified. Specify a unique machine ID with a minimum length of 1 and a maximum length of 48 characters. For example: 304238303030343700000000000. Use the **tklmMachineIdentityList** command to locate machine identities.

-machineText
Required if the value of **-machineID** or **-machineUUID** is not specified. Specify a unique, user-supplied descriptive label greater than zero bytes and not more than 96 bytes for a machine, such as myEncryptedDS5000.

Example

This Jython-formatted command adds a machine identity.

```
print AdminTask.tklmMachineIdentityAdd
('[-machineID 304238303030343700000000000]')
```

tklmMachineIdentityDelete

Use the **tklmMachineIdentityDelete** command to remove a machine identity from the IBM Security Key Lifecycle Manager data store.

Note: The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager. Use the REST interfaces instead.

Purpose

Use this command to remove a machine identity from the IBM Security Key Lifecycle Manager data store.

Permissions

Your role must have a permission to the delete action and a permission to the DS5000 device group to which the machine identity is attached.

Syntax

tklmMachineIdentityDelete -machineUUID *machineuuidvalue* **-machineID** *machineidentitystring* | **-machineText** *machineidentitytext*

Parameters

-machineID
Required if the value of **-machineText** or **-machineUUID** is not specified. Specify a unique machine ID with a minimum length of 1 and a maximum length of 48 characters. For example: 304238303030343700000000000. Use the **tklmMachineIdentityList** command to locate machine identities.

-machineText
Required if the value of **-machineID** or **-machineUUID** is not specified. Specify a

unique, user-supplied descriptive label greater than zero bytes and not more than 96 bytes for a machine, such as myEncryptedDS5000.

-machineUUID
Required if the value of **-machineText** or **-machineID** is not specified. Specify a value for a unique universal identifier for the machine, such as MACHINE-bf57ca0d-1299-4bc7-9c9c-2fa29a99c7c9. Use the **tklmMachineIdentityList** command to locate the machine uuid.

Example

This Jython-formatted command removes a machine identity from the IBM Security Key Lifecycle Manager data store.

```
print AdminTask.tklmMachineIdentityDelete
('[-machineID 3042383030303437000000000000]')
```

tklmMachineIdentityList

Use the **tklmMachineIdentityList** command to list the known machine identities for a DS5000 device group.

Note: The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager. Use the REST interfaces instead.

Purpose

Use this command to list the known machine identities for a DS5000 device group.

Permissions

Your role must have a permission to the view action and a permission to the DS5000 device group.

Syntax

tklmMachineIdentityList

Parameters

There are no parameters.

Example

This Jython-formatted command lists the machine identities that are known for a DS5000 device group.

```
print AdminTask.tklmMachineIdentityList ()
```

tklmMachineIdentityUpdate

Use the **tklmMachineIdentityUpdate** command to update the machine text of a machine in the IBM Security Key Lifecycle Manager database.

Note: The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager. Use the REST interfaces instead.

Purpose

Use this command to update the machine text of a machine in the IBM Security Key Lifecycle Manager database.

Permissions

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

Syntax

tklmMachineIdentityUpdate -machineUUID *machineidentityuuid* **-machineText** *machineidentitytext*

Parameters

-machineUUID
Required. Specify a value for a unique universal identifier for the machine, such as MACHINE-bf57ca0d-1299-4bc7-9c9c-2fa29a99c7c9. Use the **tklmMachineIdentityList** command to locate the machine uuid.

-machineText
Required. Specify a unique, user-supplied descriptive label greater than zero bytes and not more than 96 bytes, such as myEncryptedDS5000.

Example

This Jython-formatted command updates the text that describes a specific machine.

```
print AdminTask.tklmMachineIdentityUpdate
('[-machineUUID MACHINE-bf57ca0d-1299-4bc7-9c9c-2fa29a99c7c9
-machineText myEncryptedDevice]')
```

tklmPendingClientCertAccept

Use the **tklmPendingClientCertAccept** command to accept and also trust a pending client certificate of type SSLCLIENT. This command also puts the certificate into the keystore, marking the certificate as trusted to allow the client to initiate secure communication with the IBM Security Key Lifecycle Manager server.

Note: The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager. Use the REST interfaces instead.

Purpose

Use this command to accept and also trust a pending KMIP client certificate of type SSLCLIENT. This command also puts the certificate into the keystore, marking the certificate as trusted to allow the client to initiate secure communication with the IBM Security Key Lifecycle Manager server.

For example, you might set the value of the **enableClientCertPush** property to `true` and then run the **tklmPendingClientCertAccept** command to place a pending client certificate in the keystore.

To enable pending certificates to be added from a valid secure socket connection, you must previously configure SSL on the IBM Security Key Lifecycle Manager server. Otherwise, no pending certificate action is taken.

Permissions

Your role must have a permission to the configure action.

Syntax

tklmPendingClientCertAccept -uuid *certificateUUID* **-alias** *certalias*

Parameters

alias
Required. Specify a unique name for the new certificate. The name is not case-sensitive. If you specify MY Cert1, the value is stored as my cert1.

Note: Do not use a value such as aaa00000000000000002 where the value begins with three alphabetic characters followed by 18 numeric characters. IBM Security Key Lifecycle Manager uses this format to generate a key group with symmetric keys.

Do not use forward slash (/) or backslash (\) characters in the value.

uuid
Required. Specifies the Universal Unique Identifier of the certificate.

Example

This Jython-formatted command accepts a pending SSLCLIENT client certificate.

```
print AdminTask.tklmPendingClientCertAccept
('[-uuid CERTIFICATE-6a577437-e725-48bf-a836-00a47df64e78 -alias mycert1]')
```

tklmPendingClientCertList

Use the **tklmPendingClientCertList** command to list pending client certificates of type SSLCLIENT.

Note: The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager. Use the REST interfaces instead.

Purpose

Use this command to list pending client certificates of type SSLCLIENT.

Permissions

Your role must have a permission to the view action.

Syntax

tklmPendingClientCertList -v

Parameters

-v [y | n]
Optional. Verbose. The default is n, or no extra information. To list more information about a certificate, specify:

```
-v y
```

Example

This Jython-formatted command lists all of the pending client certificates.

```
print AdminTask.tklmPendingClientCertList()
```

tklmPendingClientCertReject

Use the **tklmPendingClientCertReject** command to reject a pending client certificate of type SSLCLIENT. This command also discards the certificate data from the IBM Security Key Lifecycle Manager database. The certificate cannot be used for secure communication.

Note: The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager. Use the REST interfaces instead.

Purpose

Use this command to reject a pending client certificate of type SSLCLIENT. This command also discards the certificate data from the IBM Security Key Lifecycle Manager database. The certificate cannot be used for secure communication.

If you reject a certificate and the device reestablishes a connection to IBM Security Key Lifecycle Manager, the certificate is again added to the pending list.

Permissions

Your role must have a permission to the configure action.

Syntax

tklmPendingClientCertReject -uuid *certificateUUID*

Parameters

uuid

Required. Specifies the Universal Unique Identifier of the certificate.

Example

This Jython-formatted command rejects a pending client certificate.

```
print AdminTask.tklmPendingClientCertReject
('[-uuid CERTIFICATE-6a577437-e725-48bf-a836-00a47df64e78]')
```

tklmPendingDeviceAccept

Use the **tklmPendingDeviceAccept** command to accept a device from a pending list. After acceptance from the pending list, IBM Security Key Lifecycle Manager serves keys to the device upon request.

Note: The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager. Use the REST interfaces instead.

Purpose

Use this command to accept a device from a pending list. After acceptance from the pending list, IBM Security Key Lifecycle Manager serves keys to the device upon request.

This command does not store any relationships that a device might have with a specific machine in the DS5000 device family. You can only accept devices that you have permission to create in a device group.

Permissions

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

Syntax

tklmPendingDeviceAccept -deviceUUID *deviceuniversalid* **-usage** {LTO | 3592 | DS5000 | DS8000 | BRCD_ENCRYPTOR | ONESECURE | GENERIC | *userdevicegroup*} **-deviceText** *devicetextvalue*

Parameters

`-deviceText`
Optional. Specifies unique text greater than with a minimum length greater than zero bytes and a maximum length of 96 bytes that describes a DS5000 storage server. Use the **`tklmDeviceUpdate`** command to update this value.

`-deviceUUID`
Required. Specify a value for a unique universal identifier for the device, such as `DEVICE-7d588437-e725-48bf-a836-00a47df64e78`. Use the **`tklmPendingDeviceList`** or the **`tklmPendingMachineDeviceList`** command to locate the pending device uuid.

`-usage`
Optional. Specify a device group, such as DS5000. You can include the following values:

LTO Specifies the LTO device group.

3592 Specifies the 3592 device group.

DS5000
Specifies the DS5000 device group.

DS8000
Specifies the DS8000 device group.

BRCD_ENCRYPTOR
Specifies the BRCD_ENCRYPTOR device group that is in the LTO device family.

ONESECURE
Specifies the ONESECURE device group that is in the DS5000 device family.

GENERIC
Specifies a device family that uses the Key Management Interoperability Protocol to interact with IBM Security Key Lifecycle Manager. The GENERIC device group enables management of KMIP objects.

Do not use the command-line interface to add a device to the GENERIC device group, or to change a GENERIC device group attribute.

userdevicegroup
: Specifies a user-defined group that is based on a supported device family.

Example

This Jython-formatted command accepts a specific device in the pending list.

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

tklmPendingDeviceList

Use the **`tklmPendingDeviceList`** command to list all of the pending devices that are associated with a specific device group, or all devices if no device group is specified.

Note: The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager. Use the REST interfaces instead.

Purpose

Use this command to list all of the pending devices that are associated with a specific device group, or all devices if no device group is specified. You can only list devices for device groups that you have permissions to view.

Permissions

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

Syntax

tklmPendingDeviceList -usage {LTO | 3592 | DS5000 | DS8000 }

Parameters

`-usage`
: Optional. Specify a device group such as 3592. You can include the following values:

LTO Specifies the LTO device group.

3592 Specifies the 3592 device group.

DS5000
: Specifies the DS5000 device group.

DS8000
: Specifies the DS8000 device group.

Example

This Jython-formatted command lists all of the 3592 pending devices.

```
print AdminTask.tklmPendingDeviceList ('[-usage 3592]')
```

tklmPendingDeviceReject

Use the **`tklmPendingDeviceReject`** command to reject a device from a list of pending devices.

Note: The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager. Use the REST interfaces instead.

Purpose

Use this command to reject a device from a list of pending devices. You can only reject devices for device groups that you have permissions to create.

Permissions

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

Syntax

tklmPendingDeviceReject -deviceUUID *deviceuniversalid*

Parameters

`-deviceUUID`

Required. Specify a value for a unique universal identifier for the device, such as `DEVICE-7d588437-e725-48bf-a836-00a47df64e78`. Use the **`tklmPendingDeviceList`** or the **`tklmPendingMachineDeviceList`** command to locate the pending device uuid.

Example

This Jython-formatted command rejects a device from the list of pending devices.

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

tklmPendingMachineDeviceAccept

Use the **`tklmPendingMachineDeviceAccept`** command to accept a pending machine-to-device relationship. The command also adds the pending device to the IBM Security Key Lifecycle Manager database.

Note: The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager. Use the REST interfaces instead.

Purpose

Use this command to accept a pending machine-to-device relationship. The command also adds the pending device to the IBM Security Key Lifecycle Manager database.

This command does not specify that IBM Security Key Lifecycle Manager serves keys to the device upon request.

Permissions

Your role must have a permission to the create action and a permission to the DS5000 device group.

Syntax

tklmPendingMachineDeviceAccept { **-deviceUUID** *deviceuuid* **-machineID** *machineidentitystring* | **-machineText** *machineidentitytext* }

Parameters

-deviceUUID
Required. Specify a value for a unique universal identifier for the device, such as DEVICE-7d588437-e725-48bf-a836-00a47df64e78. Use the **tklmPendingDeviceList** or the **tklmPendingMachineDeviceList** command to locate the pending device uuid.

-machineID
Required if the value of **-machineText** is not specified. Specify a unique machine ID with a minimum length of 1 and a maximum length of 48 characters. For example: 30423830303034370000000000000. Use the **tklmPendingMachineDeviceList** command to locate pending machine identities.

-machineText
Required if the value of **-machineID** is not specified. Specify a unique, user-supplied descriptive label greater than zero bytes and not more than 96 bytes for a machine, such as myEncryptedDS5000.

Example

This Jython-formatted command accepts the relationship of a specific device and machine.

```
print AdminTask.tklmPendingMachineDeviceAccept
('[-deviceUUID DEVICE-7d588437-e725-48bf-a836-00a47df64e78
-machineID 30423830303034370000000000000]')
```

tklmPendingMachineDeviceList

Use the **tklmPendingMachineDeviceList** command to list all of the pending devices that are associated with a specific machine ID, or all pending devices, if no machine ID is specified.

Note: The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager. Use the REST interfaces instead.

Purpose

Use this command to list all of the pending devices that are associated with a specific machine ID, or all pending devices, if no machine ID is specified.

Permissions

Your role must have a permission to the view action and a permission to the DS5000 device group.

Syntax

tklmPendingMachineDeviceList -machineID *machineidentitystring* **-machineText** *machineidentitytext*

Parameters

-machineID
Optional. Specify a unique machine ID, which is a concatenation of the Worldwide name and the volume serial number. The value has a minimum length of 1 and a maximum length of 48 characters. For example, specify 3042383030303437000000000000. To find a machine identity, use the **tklmMachineIdentityList** command.

-machineText
Optional. Specify a unique, user-supplied descriptive label greater than zero bytes and not more than 96 bytes for a machine, such as myEncryptedDS5000.

Example

This Jython-formatted command lists all of the pending devices that are associated with a specific machine ID.

```
print AdminTask.tklmPendingMachineDeviceList
('[-machineID 3042383030303437000000000000]')
```

tklmPendingMachineDeviceReject

Use the **tklmPendingMachineDeviceReject** command to reject a pending relationship of a device and machine from a list of pending requests. Rejection also deletes the pending relationship of the device to a machine identifier in the IBM Security Key Lifecycle Manager database.

Note: The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager. Use the REST interfaces instead.

Purpose

Use this command to reject a pending relationship of a device and machine from a list of pending requests. Rejection also deletes the pending relationship of the device to a machine identifier in the IBM Security Key Lifecycle Manager database.

Permissions

Your role must have a permission to the *create* action and a permission to the DS5000 device group

Syntax

tklmPendingMachineDeviceReject -deviceUUID *deviceuuid* **-machineUUID** *machineidentityuuid*

Parameters

-deviceUUID
Required. Specify a value for a unique universal identifier for the device, such as DEVICE-7d588437-e725-48bf-a836-00a47df64e78. Use the **tklmDeviceList** command to locate the device uuid.

-machineUUID
Required. Specify a value for a unique universal identifier for the machine, such as MACHINE-bf57ca0d-1299-4bc7-9c9c-2fa29a99c7c9. Use the **tklmMachineIdentityList** command to locate the machine uuid.

Example

This Jython-formatted command rejects the pending relationship of a device to a machine ID from the pending list and also deletes the pending relationship in the IBM Security Key Lifecycle Manager database.

```
print AdminTask.tklmPendingMachineDeviceReject
('[-deviceUUID DEVICE-7d588437-e725-48bf-a836-00a47df64e78
 -machineUUID MACHINE-bf57ca0d-1299-4bc7-9c9c-2fa29a99c7c9]')
```

tklmReplicationConfigDeleteEntry

Use the **tklmReplicationConfigDeleteEntry** command to delete a property in the IBM Security Key Lifecycle Manager replication configuration file.

Note: The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager. Use the REST interfaces instead.

Purpose

Use this parameter to delete a property in the IBM Security Key Lifecycle Manager replication configuration file.

Permissions

Your role must have a permission to the configure action.

Syntax

tklmReplicationConfigDeleteEntry -name *propertyname*

Parameters

-name
Required. Specify property name in the IBM Security Key Lifecycle Manager replication configuration file.

Example

This Jython-formatted command deletes the **backup.ClientPort4** parameter from the replication configuration file.

```
print AdminTask.tklmReplicationConfigDeleteEntry('[-name backup.ClientPort4]')
```

tklmReplicationConfigGetEntry

Use the **tklmReplicationConfigGetEntry** command to return the current value or values of a property in the IBM Security Key Lifecycle Manager server replication configuration file.

Note: The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager. Use the REST interfaces instead.

Purpose

Use this parameter to return the current value or values of a property in the IBM Security Key Lifecycle Manager replication configuration file.

Permissions

Your role must have a permission to the configure action.

Syntax

tklmReplicationConfigGetEntry -name *propertyname*

Parameters

-name
Required. Specify the name of the property in the IBM Security Key Lifecycle Manager replication configuration file.

Example

This Jython-formatted command obtains the current value of the **backup.ClientPort4** parameter from the ReplicationSKLMgrConfig.properties replication configuration file.

```
print AdminTask.tklmReplicationConfigGetEntry('[-name backup.ClientPort4]')
```

tklmReplicationConfigList

Use the **tklmReplicationConfigList** command to list the contents of the IBM Security Key Lifecycle Manager replication configuration file.

Note: The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager. Use the REST interfaces instead.

Purpose

Use this parameter to list the contents of the IBM Security Key Lifecycle Manager replication configuration file.

Permissions

You must have a valid permission to an action such as Configure, or a valid permission to an action and a device group.

Syntax

tklmReplicationConfigList -v {y| n}

Parameters

There are no required parameters.

-v [y | n]
Verbose. The default is n. This command returns verbose information, regardless of the verbose setting.

Example

This Jython-formatted command verbosely lists the contents of the IBM Security Key Lifecycle Manager replication configuration file.

```
print AdminTask.tklmReplicationConfigList()
```

tklmReplicationConfigUpdateEntry

Use the **tklmReplicationConfigUpdateEntry** command to change an existing entry or to add an entry in the IBM Security Key Lifecycle Manager replication configuration file.

Note: The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager. Use the REST interfaces instead.

Purpose

Use the **tklmReplicationConfigUpdateEntry** command to change an existing entry or to add an entry in the IBM Security Key Lifecycle Manager replication configuration file.

Permissions

Your role must have a permission to the configure action.

Syntax

tklmReplicationConfigUpdateEntry -name *propertyname* **-value** *propertyvalue*

Parameters

-name
Required. Specify the name of the property.

-value
Required. Specify the value of the property.

Examples

This Jython-formatted command updates the **backup.CheckFrequency** entry in the IBM Security Key Lifecycle Manager replication configuration file to 60.

```
print AdminTask.tklmReplicationConfigUpdateEntry('[-name backup.CheckFrequency
-value 60]')
```

tklmReplicationNow

Use the **tklmReplicationNow** command to immediately run IBM Security Key Lifecycle Manager replication and forces a backup to be sent to the configured clones.

Note: The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager. Use the REST interfaces instead.

Purpose

Use this command to immediately run IBM Security Key Lifecycle Manager replication and to force a backup to be sent to the configured clones.

Syntax

tklmReplicationNow -hostname *hostname* | **-port** *portnum*

Parameters

Note: If either host name or port parameter is coded, the other must be too.

-hostname
Optional. Specify the host to replicate to.

-port
Optional. Specify the port to connect to the replication clone through.

Example

This Jython-formatted command replicates IBM Security Key Lifecycle Manager to all clones defined in the `ReplicationSKLMgrConfig.properties` replication configuration file.

```
print AdminTask.tklmReplicationNow()
```

The following command replicates IBM Security Key Lifecycle Manager to a specific server.

```
print AdminTask.tklmReplicationNow('[-hostname myserver -port 1111]')
```

tklmReplicationStart

Use the **tklmReplicationStart** command to start the IBM Security Key Lifecycle Manager replication task.

Note: The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager. Use the REST interfaces instead.

Purpose

Use this command to start the IBM Security Key Lifecycle Manager replication task.

Syntax

tklmReplicationStart

Parameters

There are no parameters.

Example

This Jython-formatted command starts the IBM Security Key Lifecycle Manager replication task.

```
print AdminTask.tklmReplicationStart()
```

tklmReplicationStatus

Use the **tklmReplicationStatus** command to obtain information about the IBM Security Key Lifecycle Manager replication task such as operational status and replication schedules.

Note: The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager. Use the REST interfaces instead.

Purpose

Use this command to obtain information about the IBM Security Key Lifecycle Manager replication task such as operational status and replication schedules.

Syntax

tklmReplicationStatus

Parameters

There are no parameters.

Example

This Jython-formatted command displays status information about the IBM Security Key Lifecycle Manager replication task.

```
print AdminTask.tklmReplicationStatus()
```

tklmReplicationStop

Use the **tklmReplicationStop** command to stop the IBM Security Key Lifecycle Manager replication task.

Note: The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager. Use the REST interfaces instead.

Purpose

Use this command to stop the IBM Security Key Lifecycle Manager replication task.

Syntax

tklmReplicationStop -confirm {y| n}

Parameters

-confirm [y | n]
Required. Confirms that stop must occur. The default is n.

Example

This Jython-formatted command stops the IBM Security Key Lifecycle Manager replication task.

```
print AdminTask.tklmReplicationStop('[-confirm y]')
```

tklmSecretDataDelete

Use the **tklmSecretDataDelete** command to delete secret data. For example, you might delete the secret data that is registered by the client.

Note: The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager. Use the REST interfaces instead.

Purpose

Use this command to delete secret data. For example, you might delete the secret data that is registered by the client.

Permissions

You must have the klmSecurityOfficer role.

Syntax

tklmSecretDataDelete -uuid *secretdatauuid*

Parameters

-uuid
Required. Specify the unique universal identifier of an existing secret data. For example: K_SEC_DATA-d3ee4491-f96e-495d-bb37-fc03748924ba

Example

This Jython-formatted command deletes secret data.

```
print AdminTask.tklmSecretDataDelete
 ('[-uuid K_SEC_DATA-d3ee4491-f96e-495d-bb37-fc03748924ba]')
```

tklmSecretDataList

Use the **tklmSecretDataList** command to list secret data. For example, you might list the secret data that is registered by the client.

Note: The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager. Use the REST interfaces instead.

Purpose

Use this command to list secret data that IBM Security Key Lifecycle Manager provides. For example, you might list the secret data that is registered by the client.

Permissions

You must have the `klmSecurityOfficer` role.

Syntax

tklmSecretDataList -uuid *secretdatauuid* **-v** {y| n}

Parameters

`-uuid`
Optional. Specify the unique universal identifier of existing secret data. For example: `K_SEC_DATA-d3ee4491-f96e-495d-bb37-fc03748924ba`

`-v [y | n]`
Optional. Verbose. The default is `n`, or no extra information other than the uuid and name. To list more information, specify:

```
-v y
```

Example

This Jython-formatted command verbosely lists secret data.

```
print AdminTask.tklmSecretDataList
 ('[-uuid K_TEMPLATE-d3ee4491-f96e-495d-bb37-fc03748924ba -v y]')
```

The output is generated as shown in the example:

```
Unique Identifier = K_SEC_DATA-d3ee4491-f96e-495d-bb37-fc03748924ba
Cryptographic Length = 6
Operation Policy Name = default policy for secret data
Activation Date = 2009-12-01
Deactivation Date = 2010-12-01
Contact Information = 1540 Scenic Ave, Costa Mesa, CA 92626
Revocation Reason = [[TYPE PRIVILEGE_WITHDRAWN] [MESSAGE no longer used]]
Name = [[INDEX 1] [TYPE TEXT] [VALUE secret data for db2 password]],
[[INDEX 2] [TYPE URI] [VALUE secret data for xyz]]
Cryptographic Parameters = [[INDEX 1] [MODE CBC] [PAD ISO_10126]
[HASH SHA256] [ROLE ZMK]]
Object Group = [[INDEX 1] [VALUE object group 1]], [[INDEX 2] [VALUE object group2]]
Link =
Digest = [[INDEX 0] [HASH SHA256] [VALUE digest value of SHA256]],
[[INDEX 1] [HASH SHA1] [VALUE digest value of SHA1]]
Application Specific Information = [[INDEX 1] [NAMESPACE https://www.ibm.com/tklm]
[INFO used for tklm]],[[ INDEX 2] [NAMESPACE https://www.ibm.com/tklm1]
[INFO used for tklm1]]
Custom Attribute = [[NAME x-clientattr1] [[INDEX 0] [TYPE TEXT] [VALUE value1]]],
[[NAME x-clientattr2] [[[INDEX 0] [TYPE TEXT] [VALUE value2]]], [[NAME y-serverattr1]
[[[INDEX 0] [TYPE TEXT] [VALUE value3]], [[INDEX 1] [VALUE value4]]]]
Type = PASSWORD
Initial Date = 2009-12-01
Last Change Date = 2009-12-01
Destroy Date =
Archive Date =
Compromise Occurrence Date = 2010-06-01
Compromise Date =
Lease Time =
State =
Owner = base64 encoded SHA-256 hash public key of client certificate
```

tklmSecretKeyCreate

Use the **tklmSecretKeyCreate** command to create one or more symmetric keys.

Note: The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager. Use the REST interfaces instead.

Purpose

Use this command to create one or more symmetric keys.

`Do not use other key-generating tools such as keytool or the iKeyman utility` to create or to modify keys or certificates. Use IBM Security Key Lifecycle Manager.

Permissions

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

Syntax

tklmSecretKeyCreate -alias *keyalias* **-aliasRange** *prefixhexnumber1-hexnumberN* **-keyStoreName** *keystorename* **-keyGroupUuid** *keygroupname* **-usage** {LTO | DS5000 | BRCD_ENCRYPTOR | ONESECURE | GENERIC | *userdevicegroup*} **-numOfKeys** *integernumberofkeys*

Parameters

You must specify a value for either the **-alias** or the **-aliasRange** parameter.

-alias

Required if you do not specify the **-aliasRange** parameter. Specify a key alias.

The value for the alias cannot exceed 12 characters. If you specify a value for the **-alias** parameter, also specify a value for the **-numOfKeys** parameter.

For example:

```
-alias abc
```

Note: Do not specify a value that is an alias range, or a value that contains a hyphen or dash character (-), which is reserved for use in the value for an alias range.

-aliasRange

Required if a value is not specified for the **-alias** parameter. Specify a three character common alias prefix, followed by a range of numbers for a set with multiple keys.

For example:

```
-aliasRange ibm1-a
```

Alternatively, you might specify:

```
-aliasRange xyz01-fff
```

-keyGroupUuid

Specify the uuid of the key group. For example, `GROUP-7d588437-e725-48bf-a836-00a47df64e78`.

-keyStoreName
Required. Specify the name of the keystore.

-numOfKeys
Specify the number of keys to create. If you specify a value for the **-alias** parameter, also specify a value for the **-numOfKeys** parameter. The value must be a positive integer not greater than 9999.

-usage
Required. Specify the target application usage.

You can include the following values:

LTO Specifies the LTO device group.

DS5000
Specifies the DS5000 device group.

GENERIC
Specifies a device family that uses the Key Management Interoperability Protocol to interact with IBM Security Key Lifecycle Manager. The GENERIC device group enables management of KMIP objects.

Do not use the command-line interface to add a device to the GENERIC device group, or to change a GENERIC device group attribute.

Example

This Jython-formatted command creates 10 symmetric keys for use with LTO tape drives.

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

This Jython-formatted command creates 10 symmetric keys in a keystore with a name that includes spaces.

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

This Jython-formatted command creates a range of symmetric keys.

```
print AdminTask.tklmSecretKeyCreate ('[-aliasrange xyz1-f
-keyStoreName defaultKeyStore
 -usage LTO
  -keyGroupUuid KEYGROUP-316408ac-f433-4c11-92bc-0de46d05bee9]')
```

tklmServedDataList

Use the `tklmServedDataList` command to query the database and list the served key data. For example, you might list which devices were served a specific key, or list the keys that were served to a specific device. To prevent running out of memory if all audits are returned, specify a time interval for a list of audits.

Note: The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager. Use the REST interfaces instead.

Purpose

Use this command to query the database and list the served key data. For example, you might list which devices were served a specific key, or list the keys that were served to a specific device. To prevent running out of memory if all audits are returned, specify a time interval for a list of audits.

Permissions

Your role must have a permission to the audit action.

Syntax

tklmServedDataList [-attributeName {alias1 | alias2 | dki} **-attributeValue** *serveddatavalue* **-volser** *volumeserialnumber* **-dateBefore** *YYYY-MM-DD* **-dateAfter** *YYYY-MM-DD* **-usage** {LTO | 3592 | DS5000 | DS8000 | BRCD_ENCRYPTOR | ONESECURE | *userdevicegroup*}]

Parameters

There are no required parameters. Specify either -attributeName and -attributeValue to list which devices were served a specific key, or specify -volser for information about which keys were served to a specific device.

`-attributeName`
Specify the type of served data:

alias1 Specifies a default alias for a certificate that is used by a 3592 tape drive or a DS8000 Turbo drive. Not used for an LTO tape drive or DS5000 storage server.

- 3592 tape drive

  The value is optional for a 3592 tape drive and specifies the primary certificate that the device in the 3592 device family uses if the primary certificate is not available. If this attribute is not specified, the partner default certificate is used, as specified by a table entry for the device group in the IBM Security Key Lifecycle Manager database.
- DS8000 Turbo drive

  The value is optional for a DS8000 Turbo drive and matches the label **Primary certificate for image** in the graphical user interface panels for a DS8000 Turbo drive.

Use the **`tklmDeviceGroupAttributeList`** and **`tklmDeviceGroupAttributeUpdate`** commands to view and change the value. This value was previously stored in the obsolete configuration parameter **`drive.default.alias1`**.

alias2 Used for a 3592 tape drive or a DS8000 Turbo drive. Not used for an LTO tape drive or DS5000 storage server.

- 3592 tape drive

  This attribute specifies a default alternative alias for a 3592 tape drive. This value can be the same, or different from the value that is specified for the primary certificate.

  The value specifies the secondary certificate that the device in the 3592 device family uses if the primary certificate is not available. If this attribute is not specified, the partner default certificate is used, as specified by a table entry for the device group in the IBM Security Key Lifecycle Manager database.

- DS8000 Turbo drive

  For a device in the DS8000 device family, the value specifies a secondary certificate that is available for use. For example, you might use this certificate to unlock a DS8000 Turbo drive in the case of a deadlock condition.

Use the **tklmDeviceGroupAttributeList** and **tklmDeviceGroupAttributeUpdate** commands to view and change the value. This value was previously stored in the obsolete configuration parameter **drive.default.alias2**.

dki Data key identifier, used only for an LTO tape drive.

-attributeValue
Identifies the served data. For example, if -attributeName is alias1, then -attributeValue might be cert1.

-dateBefore *YYYY-MM-DD*
If this date is the only date specified, list the audits that are made before this date. Hyphens are required in the date value.

To list audits that are made between the before and after dates, specify both values.

-dateAfter *YYYY-MM-DD*
If this date is the only date that is specified, list the audits that are made after this date. Hyphens are required in the date value.

To list audits that are made between the before and after dates, specify both values.

-usage
Specify an existing device group, such as myLTO.

You can include the following values:

LTO Specifies the LTO device group.

DS5000
Specifies the DS5000 device group.

GENERIC
Specifies a device family that uses the Key Management Interoperability Protocol to interact with IBM Security Key Lifecycle Manager. The GENERIC device group enables management of KMIP objects.

Do not use the command-line interface to add a device to the GENERIC device group, or to change a GENERIC device group attribute.

-volser
Specify the volume and serial number of a tape cartridge. For example, specify TEST.

-kmipClientCertUUID
Specify UUID of the KMIP client certificate.

-serialNumber
Specify the device serial number.

Examples

This Jython-formatted command returns a list of all devices that were served a certificate that is identified as cert1.

```
print AdminTask.tklmServedDataList \
 ('[-attributeName alias1 -attributeValue cert1 -dateBefore 2010-06-11]')
```

This Jython-formatted command returns a list of all keys served to a device identified as TEST.

```
print AdminTask.tklmServedDataList \
 ('[-volser TEST -dateBefore 20100611 -dateAfter 2010-11-30]')
```

tklmTrustStoreCertAdd

Use the **tklmTrustStoreCertAdd** command to add a certificate from a certificate file that is in either DER or base64 format to the IBM Security Key Lifecycle Manager internal truststore.

Note: The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager. Use the REST interfaces instead.

Purpose

Use this command to add a certificate from a certificate file that is in either DER or base64 format to the IBM Security Key Lifecycle Manager internal truststore.

Using the certificate allows communication between IBM Security Key Lifecycle Manager and the device that identifies itself by using this certificate or the root certificate for this certificate.

Permissions

Your user ID must have the klmSecurityOfficer role.

Syntax

tklmTrustStoreCertAdd -fileName *pathandfilename* **-format** {base64 | DER} **-alias** *certalias*

Parameters

-fileName
Required. Specifies the file name to import containing the certificate data. The imported file is stored in a keystore location relative to the *SKLM_HOME* directory.

-format
Specify either base64 (default, if this parameter is not specified) or DER (Distinguished Encoding Rules) format.

-alias
Required. Specify a unique name for the certificate.

Example

This Jython-formatted command adds a certificate to a certificate file that is in DER format to the IBM Security Key Lifecycle Manager internal truststore.

```
print AdminTask.tklmTrustStoreCertAdd
 ('[-fileName d:\\mypath\\mycertfilename.der
  -format DER -alias myCertAlias]')
```

tklmTrustStoreCertDelete

Use the **`tklmTrustStoreCertDelete`** command to delete a certificate that is in the IBM Security Key Lifecycle Manager internal truststore.

Note: The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager. Use the REST interfaces instead.

Purpose

Use this command to delete a certificate that is in the IBM Security Key Lifecycle Manager internal truststore.

Permissions

Your user ID must have the klmSecurityOfficer role.

Syntax

tklmTrustStoreCertDelete -alias *certalias*

Parameters

`-alias`
Required. Specify a unique name for the certificate.

Example

This Jython-formatted command deletes a certificate that is in the IBM Security Key Lifecycle Manager internal truststore.

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

tklmTrustStoreCertList

Use the **`tklmTrustStoreCertList`** command to list certificates that are the IBM Security Key Lifecycle Manager internal truststore.

Note: The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager. Use the REST interfaces instead.

Purpose

Use this command to list certificates that are the IBM Security Key Lifecycle Manager internal truststore.

Permissions

Your user ID must have the `klmSecurityOfficer` role.

Syntax

tklmTrustStoreCertList -alias *certalias* **-v** {y| n}

Parameters

-alias
Optional. Specify a unique name for the certificate. To list all certificates, do not specify an alias.

-v [y | n]
Optional. Verbose. The default is `n`, or no extra information. To list more information about a certificate, specify:

```
-v y
```

Example

This Jython-formatted command lists all certificates that are in the IBM Security Key Lifecycle Manager internal truststore.

```
print AdminTask.tklmTrustStoreCertList()
```

tklmVersionInfo

Use the **tklmVersionInfo** command to list version numbers for IBM Security Key Lifecycle Manager and associated middleware.

Note:

- The **tklmVersionInfo** command provides the information about DB2 version that is actually the service level information of DB2 that IBM Security Key Lifecycle Manager uses. However, the service level information for DB2 is shown differently on Windows and Linux operating systems. For example, for a DB2 version 10.1.0.2, the version is shown as 10.1.200.38 on Windows operating system. On Linux operating system, the version is shown as 10.1.0.2.
- The IBM Security Key Lifecycle Manager command-line interface commands will be deprecated in the later versions of IBM Security Key Lifecycle Manager.

Purpose

Use this command to list version numbers for IBM Security Key Lifecycle Manager and associated middleware.

Permissions

Your role must have a permission to the configure action.

Syntax

tklmVersionInfo

Parameters

There are no parameters.

Example

This Jython-formatted command lists version numbers for IBM Security Key Lifecycle Manager and associated middleware.

```
print AdminTask.tklmVersionInfo()
```

Server configuration properties and database values

IBM Security Key Lifecycle Manager uses both properties in the `SKLMConfig.properties` file and values in the IBM Security Key Lifecycle Manager database to control operation of the IBM Security Key Lifecycle Manager server. Some properties are not present in the `SKLMConfig.properties` file until a runtime task creates the property and its value.

The server configuration properties are either in the `*SKLM_HOME*/config/SKLMConfig.properties` file or are values in the IBM Security Key Lifecycle Manager database. Your role must have the configure permission (`klmConfigure`) to read or change an IBM Security Key Lifecycle Manager property.

The order of property settings in the `SKLMConfig.properties` file does not matter. Comments can appear in the file. To add a comment, use a " #" in the first column of a line.

Note: If you use the graphical user interface, or command-line interface, you can change the IBM Security Key Lifecycle Manager configuration file while the server is running. Depending on the change, you might see a message that indicates you to restart the IBM Security Key Lifecycle Manager server for the change to take effect.

Syntax statements in these properties descriptions use the following conventions:

| Specify not more than one of the items.

[ ] Specify optional items. A comma (,) is a valid separator between items.

{ } Repeat 0 or more times.

underline
Indicates the default value. For example, <u>success</u>.

space before and after equals (=) sign
A space is permitted before and after the equals sign. These syntax examples omit a space.

Audit.event.outcome

Specifies whether events are occurring as a result of successful operations, unsuccessful operations, or both are audited.

Note: Do not delete this property from the properties file.

Audit.event.outcome={*outcome*[,*outcome*]}
Specify **success** for events to be logged which occur as a result of successful operations. Specify `failure` for events to be logged which occur as a result of unsuccessful operations. Only audit events that resulted in the specified outcome are recorded.

Required
Yes.

Values
<u>success</u> | <u>failure</u>

Both can be specified separated by comma or semicolon.

Default
success, failure

Examples

Specification for this configuration value is shown in the following example.

```
Audit.event.outcome=failure
```

To enable both successful and unsuccessful cases:

```
Audit.event.outcome=success,failure
```

Audit.eventQueue.max

This property sets the maximum number of event objects to be held in the memory queue before they are flushed to file.

Audit.eventQueue.max=*numberOfEvents*

This property is optional but suggested. The default is zero.

Required

Optional. Suggested.

Values

0 - *numberOfEvents*

A value of zero means flush immediately. *numberOfEvents* is an integer greater than zero.)

Default

0

Note: A default value of zero causes all events to be written to a file.

Example

```
Audit.eventQueue.max=0
```

To avoid first failure data loss, *do not change* the queue maximum to a value other than zero.

Audit.event.types

This property specifies which audit types are sent to the audit log.

Audit.event.types={*type*[,*type*]}

Only audit events that resulted in the specified outcome are recorded.

Required

Yes.

Values

all | runtime| authentication | authentication_terminate | authorization | configuration_management | resource_management | key_management | none

Multiple values can be specified, separated by a comma or semicolon.

Note: Do not specify a value of none in combination with other values.

all All event types.

authentication

Authentication events.

authentication_terminate
Events that occur when the user logs out. Not used.

authorization
Authorization events.

configuration_management
Configuration events.

key_management
Events when changes occur in the configuration of keys.

none No events.

Note: Do not specify a value of `none` in combination with other values.

resource_management
Events when changes occur in the configuration of resources such as tape drives to the IBM Security Key Lifecycle Manager server.

runtime
Events that occur as a part of processing operations and requests that are sent to IBM Security Key Lifecycle Manager.

Default
authorization, authorization_terminate, resource_management, runtime, key_management

Examples
To collect all auditable event data, enter:

```
Audit.event.types=all
```

Another example is:

```
Audit.event.types=authorization,runtime,resource_management
```

Audit.handler.file.name

This property specifies the path and file name to which audit entries are logged, such as `sklm_audit.log`.

Audit.handler.file.name=*path/fileName*
This is the path and file name that is used as the base name in creating audit log files in the specified audit directory, which is relative to the *`SKLM_HOME`* directory.

Required
Yes.

Default
`logs/audit/sklm_audit.log`

Example
To set the base name to `my_sklm_audit.log`, enter:

```
Audit.handler.file.name=my_sklm_audit.log
```

The name of the audit log file that reaches its maximum size appends a value for the time at which the file was closed. For example, if **`Audit.handler.file.name`** value is set to `sklm_audit.log`, a closed file has a name like `sklm_audit.log.2315003554`. Higher number values indicate newer audit log files.

Audit.handler.file.size

This property specifies the size in KB that the `Audit.handler.file.name` file grows before a new file is created.

Audit.handler.file.size=*sizeInKBytes*

The value indicates the size limit at which an audit file is closed and a new audit file is written. The actual size of the resulting audit file might exceed this value by several bytes because the file is closed after the size limit is exceeded.

Required

Optional. Recommended.

Values

0 - *integervalue* (in KB).

The value of *integervalue* is a positive integer. Specifying zero sets the file size to the default value.

Default

10000

The default is 10 MB, which is 10000 KB.

Example

To set the file size to approximately 20 MB, enter:

```
Audit.handler.file.size=20000
```

Audit.handler.file.threadlifespan

This property limits the lifetime of an audit record processing thread if the value of **`audit.handler.file.multithreads`** is set to `true`.

Audit.handler.file.threadlifespan=*timeInSeconds*

This value is used during cleanup processing to allow threads to complete their work before the threads are interrupted. If a background thread is not completed its work within the life span, cleanup processing interrupts the thread.

Required

Optional.

Values

Specified in seconds.

Default

10

Example

To set the expected time to 15 seconds as the interval of time for a thread to write to the audit log, enter:

```
Audit.handler.file.threadlifespan=15
```

Audit.handler.file.multithreads

This property specifies whether the audit handler dispatches separate threads to process audit records.

Audit.handler.file.multithreads={true|false}

Specifies whether to use separate threads to process audit records.

Required

Optional.

Values

true | false

If the property is set to `true`, a separate thread is used to write the event data to the audit log, allowing the current thread of execution (operation) to continue without waiting for the write to the audit log to complete. Use of multiple threads is the default behavior.

Default

true

Example

To set multithreading to `false`, enter:

```
Audit.handler.file.multithreads=false
```

autoRestartAfterRestore

This property specifies whether to restart IBM Security Key Lifecycle Manager automatically after a successful restore operation.

autoRestartAfterRestore={true|false}

Specifies whether to restart IBM Security Key Lifecycle Manager automatically after a successful restore operation.

Required

Optional.

Values

true | false

Default

true

Example

```
autoRestartAfterRestore=true
```

backup.keycert.before.serving

This property specifies whether IBM Security Key Lifecycle Manager serves keys or certificates that are not backed up.

backup.keycert.before.serving={true|false}

A value of `true` specifies that IBM Security Key Lifecycle Manager does not serve keys or certificates that are not backed up.

Required

Optional.

Values

true | false

If `true` is specified, IBM Security Key Lifecycle Manager does not serve keys or certificates that are not backed up.

Default

true

Example

```
backup.keycert.before.serving=true
```

cert.valiDATE

This property specifies whether to carry out certificate date validation before a certificate is served.

To modify this property, you must have permissions to modify a 3592 device group or DS8000 device group.

cert.valiDATE={**true** | **false**}

A value of `true` specifies that the `notBefore` and `notAfter` dates of the certificate are used to validate the certificate. Any certificate with a date range that falls outside these Java DATE specifications cannot be used for encryption, although the certificate can still be used to decrypt, or read, encrypted tapes. If this property is set to a value of `false` or not specified, no certificate date validation is performed. This property applies to 3592 tape drives and DS8000 Turbo drives. The property is not used for LTO tape drives.

Required

Optional.

Values

true | false

On a new installation of IBM Security Key Lifecycle Manager, the value is set to `false`. If Encryption Key Manager data migration occurs, the value is set to the value specified in Encryption Key Manager. If no value is specified for Encryption Key Manager data, the value is set to the IBM Security Key Lifecycle Manager default, which is `false`.

Default

false

Example

```
cert.valiDATE=true
```

config.keystore.name

This property specifies the name of the user-provided keystore, which is determined when you first install and configure IBM Security Key Lifecycle Manager. You cannot modify the value of this property by using the command-line interface.

Note: In a running production environment, *do not modify* the keystore name. If you must modify the keystore name before production, ensure that you have a complete, current backup of your keystore.

config.keystore.name=*keystorename*

Specifies the user-provided keystore.

Required

Yes.

Default

defaultKeyStore

The default path is:

Windows

drive:\Program Files (x86)\IBM\WebSphere\AppServer\keystore

Linux, AIX®, and Solaris

path/IBM/WebSphere/AppServer/keystore

Example

`config.keystore.name=defaultKeyStore`

config.keystore.ssl.certalias

This property is an alias that points at an existing certificate that is used for SSL authentication.

Note: Do not specify a certificate for SSL use that is also specified for future use.

config.keystore.ssl.certalias=*certaliasname*

This alias points at an existing certificate that is used for SSL authentication for secure communication between a drive and IBM Security Key Lifecycle Manager.

Required

Yes.

Example

config.keystore.ssl.certalias=cert1

device.AutoPendingAutoDiscovery

Specifies whether to add a new device that contacts IBM Security Key Lifecycle Manager to a list of pending devices that you can accept or reject before key serving occurs, or to add a new device automatically to the drive table for immediate key service upon request. The attribute applies to predefined base device families and user-defined device groups.

To modify **`device.AutoPendingAutoDiscovery`**, you must have a role with permissions to modify a device group. The **`device.AutoPendingAutoDiscovery`** attribute in the IBM Security Key Lifecycle Manager database replaces the previous **`drive.acceptUnknownDrives`** and the **`ds8k.acceptUnknownDrives`** properties.

device.AutoPendingAutoDiscovery={**0** | **1**| **2**}

Specifies whether to add a device that contacts IBM Security Key Lifecycle Manager to a list of pending devices that you can accept or reject before key serving occurs, or to add a device automatically to the drive table for immediate key service upon request.

Required

Yes.

Values

0 (manual)

Both the auto pending and auto discovery functions are off. All incoming devices are rejected, and not added to the data store. You must manually add devices and machine IDs.

The corresponding choice in the graphical user interface is **Only accept manually added devices for communication.**

1 (auto accept)

The auto discovery function is on, and the auto pending function is off. All incoming devices of a valid device group are added to the data store and are automatically served keys upon request.

The corresponding choice in the graphical user interface is **Automatically accept all device requests for communication.**

Note:

- 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 can a backup.
- For all other device families, you must back up any new keys that are served.

Migrating from a previous version of IBM Security Key Lifecycle Manager sets the auto discovery value to `on` by device group if either of these conditions are true:

- `ds8k.acceptUnknownDrives=true`

  The auto discovery function is on for a device group of DS8000.
- `drive.acceptUnknownDrives=true`

  The auto discovery function is on for all valid device groups.

If both values are false, the value of **`device.AutoPendingAutoDiscovery`** is set to 0 (off). After migration, the **`ds8k.acceptUnknownDrives`** and **`drive.acceptUnknownDrives`** properties are removed from the migrated `SKLMConfig.properties` file.

Note: To allow tape drive devices of a specific type that connect to IBM Security Key Lifecycle Manager to be added and operational without an administrator validating the addition, use this setting in combination with these additional settings:

- 3592 tape drive

  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.
- LTO tape drive

  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.

2 (auto pending)

The auto pending function is on. 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.

The corresponding choice in the graphical user interface is **Hold new device requests pending my approval.**

Default

0 (off. You must manually add devices to IBM Security Key Lifecycle Manager.)

Example

```
device.AutoPendingAutoDiscovery=2
```

Suggested settings include:

Table 3. Device groups and suggested settings

Device group	Suggested value for device.AutoPendingAutoDiscovery
LTO	Any setting is acceptable if there are no device groups. However, if device groups are specified: • The auto accept option (`device.AutoPendingAutoDiscovery=`**1**) is problematic. Moving a device to another group is difficult because the keys or certificates from the family default are already served to the device. • Auto pending (`device.AutoPendingAutoDiscovery=2`) and manual (`device.AutoPendingAutoDiscovery=0`) options are better choices because an administrator has the opportunity to put the device into the correct group before keys are served.
3592	

Table 3. Device groups and suggested settings (continued)

Device group	Suggested value for device.AutoPendingAutoDiscovery
DS5000	Auto accept (`device.AutoPendingAutoDiscovery=1`) is not suggested. If auto accept is enabled, keys are generated and served to the device and administrators have no opportunity to back up the keys before data is encrypted with those keys. Auto pending (`device.AutoPendingAutoDiscovery=2`) is suggested. Keys are generated on the initial request. Before you accept the request, back up IBM Security Key Lifecycle Manager. When machine affinity is enabled for DS5000, auto pending is the easiest way to add machine identifiers to IBM Security Key Lifecycle Manager because the machine ID information is populated from the device request. There is no graphical user interface to add new machine IDs to IBM Security Key Lifecycle Manager. **Note:** If you set `device.AutoPendingAutoDiscovery=2`, a DS5000 storage server that contacts IBM Security Key Lifecycle Manager is put in the auto pending table. Before devices are accepted, a backup is suggested. However, the backup stores the devices in a pending state. If you restore this backup, devices that are previously accepted are placed in the pending table, causing requests from those devices to fail until you accept them again. Manual (`device.AutoPendingAutoDiscovery=0`) is an acceptable option unless you initially set up the system with machine affinity enabled. If you initially specify the manual setting and enable machine affinity, it is more difficult to populate the system with machine identifiers because you can only add a machine ID to IBM Security Key Lifecycle Manager by using the command-line interface. Take care to avoid errors in typing the machine ID. After you use auto pending (`device.AutoPendingAutoDiscovery=2`) to populate the system with machine IDs, changing to a manual setting is an acceptable option.
DS8000	Any setting is acceptable. In general, auto accept is the least secure setting because IBM Security Key Lifecycle Manager serves keys to any device that contacts IBM Security Key Lifecycle Manager.
GENERIC	Do not set a value. This property does not affect the GENERIC device family because devices are not supported in this family.

device.enableMachineAffinity

This database value specifies whether a specific device group is enabled to store the association of a device to an existing machine identifier in the IBM Security Key Lifecycle Manager database. An instance of the attribute is stored for each device group.

To modify this value, you must have a role with permissions to create or modify and also have permission to the DS5000 device group.

device.enableMachineAffinity={**true** | **false**}

Determines whether a specific device group is enabled to store the association of a device to an existing machine identifier in the IBM Security Key Lifecycle Manager database. An instance of the attribute is stored for each device group.

Required
Yes.

Values
true | false

true Data that records the association of a device to an existing machine identifier is stored in the IBM Security Key Lifecycle Manager database.

false No data is stored.

Default
false (off)

Example
device.enableMachineAffinity=true

disableDatabaseBackup

IBM Security Key Lifecycle Manager uses this property to disable the database backup function for IBM Security Key Lifecycle Manager on z/OS systems, on which the database must be separately backed up by the DB2® administrator.

disableDatabaseBackup={**true**|**false**}
Disables the database backup function for IBM Security Key Lifecycle Manager on z/OS systems, on which the database must be separately backed up by the DB2 administrator.

Required
Required on z/OS systems. Not present on distributed systems.

Values
true | false

The default is `true` on z/OS systems. If the property is set to `true`, database backup is disabled.

Default
true

Example
To disable database backup, enter:

```
disableDatabaseBackup=true
```

drive.acceptUnknownDrives (replaced)

This property and any value are migrated to the device.AutoPendingAutoDiscovery attribute in the IBM Security Key Lifecycle Manager database. Previously, the drive.acceptUnknownDrives property specified whether to add a LTO tape drive or 3592 tape drive that contacts IBM Security Key Lifecycle Manager to the drive table.

drive.default.alias1 (replaced)

This property is removed from the `SKLMConfig.properties` file and replaced with a default entry for the device group in the IBM Security Key Lifecycle Manager database. Use the **`tklmDeviceGroupAttributeList`** and **`tklmDeviceGroupAttributeUpdate`** commands to view and change the value. Your role must have a permission to the modify action and a permission to the appropriate device group. Previously, the property specified a default alias for a 3592 tape drive if one is not specified in the drive table. This alias specifies the system default certificate that the device uses if the device is not associated with another certificate.

drive.default.alias2 (replaced)

This property is removed from the `SKLMConfig.properties` file and replaced with a default entry for the device group in the IBM Security Key Lifecycle Manager database. Use the **`tklmDeviceGroupAttributeList`** and **`tklmDeviceGroupAttributeUpdate`** commands to view and change the value. Your role must have a permission to the modify action and a permission to the appropriate device group. Previously, the property specified a default alias for a 3592 tape drive if one was not specified in the drive table. This alias specifies the system partner certificate that the device uses if the device is not associated with another partner certificate.

ds8k.acceptUnknownDrives (replaced)

This property and any value are migrated to the device.AutoPendingAutoDiscovery attribute in the IBM Security Key Lifecycle Manager database. Previously, this property specified whether to add a DS8000 Turbo drive that contacts IBM Security Key Lifecycle Manager to the drive table.

enableClientCertPush

This property specifies whether to save pending client certificates and related data in the IBM Security Key Lifecycle Manager database.

To modify this value, you must have permission to the configure action.

enableClientCertPush={**true** | **false**}

Determines whether to use pending client certificates.

Required

Optional.

Values

true | false

true Save pending client certificates and related data.

false Do not save pending client certificates and related data.

Default

false

If no value is set, the default value is `false`.

Example

enableClientCertPush=true

fips

This property specifies whether to use the Federal Information Processing Standard.

fips={**on**|**off**}

Use the Federal Information Processing Standard.

Required

Optional.

Values
on | off

Default
off

Example
fips=on

Note: Do not use hardware-based keystore types when the fips parameter is set `on`. Setting the fips configuration parameter to `on` causes IBM Security Key Lifecycle Manager to use the IBMJCEFIPS provider for all cryptographic functions.

kmip.request.processing.hostNameLookup

This property is used to enable or disable the client host name lookup during KMIP message processing on the IBM Security Key Lifecycle Manager server.

kmip.request.processing.hostNameLookup={true|false}
When the KMIP messages are processed, host name of the client, which sends a KMIP message to the server, is written to the IBM Security Key Lifecycle Manager log for debug information. You can use this property to avoid the costly process of host name lookup.

Required
Optional

Values
true | false

Default
false

Example

```
kmip.request.processing.hostNameLookup=true
```

KMIPListener.ssl.port

This property specifies the port on which IBM Security Key Lifecycle Manager server listens for requests that communicate over the SSL socket using the Key Management Interoperability Protocol.

KMIPListener.ssl.port=*portnumber*
Port on which IBM Security Key Lifecycle Manager server listens for requests from libraries that communicate over the SSL socket using the Key Management Interoperability Protocol.

Required
Yes.

Values
An integer value that is a valid port number.

Default
The KMIP default is 5696.

Example

```
KMIPListener.ssl.port=5696
```

lock.timeout

This property specifies the interval of time in seconds that a thread for an operation waits to acquire a lock before throwing a lock exception message indicating the operation failed because dependent data is locked by another operation.

For example, some operations such as key generation require a prolonged time to create large numbers of keys. When an operation runs, it blocks other operations that require the same lock. To add or update the timeout, use **tklmConfigUpdateEntry** command.

lock.timeout=*default_value*
Specifies the interval of time in seconds that a thread for an operation waits to acquire a lock before throwing a lock exception message indicating the operation failed because dependent data is locked by another operation.

Required
No

Default (seconds)
120

maximum.keycert.expiration.period.in.years

This property sets the maximum expiration period for keys and certificate objects.

maximum.keycert.expiration.period.in.years
Set this parameter to control the maximum expiration period (in years) for keys and certificates.

Required
Optional.

Values
Integer value that is greater than zero.

Default
50

Example

```
maximum.keycert.expiration.period.in.years=25
```

maxPendingClientCerts

This property specifies the maximum number of pending client certificates allowed in the pending certificate list. If the number of pending certificates exceeds the maximum, the pending certificate and related data is not saved in the IBM Security Key Lifecycle Manager database.

To modify this value, you must have permission to the configure action.

maxPendingClientCerts=*maxcertsinteger*
Determines the maximum number of pending client certificates allowed in the pending certificate list.

Required
Optional.

Values

1 Minimum number of client certificates.

999 Maximum number of client certificates.

Default

100

If no value is set, the default value is 100.

Example

maxPendingClientCerts=200

pcache.refresh.interval

This property specifies the interval of time in minutes at which IBM Security Key Lifecycle Manager refreshes the permission cache for role-based access.

pcache.refresh.interval=*default_value*

Specifies the interval of time in minutes at which IBM Security Key Lifecycle Manager refreshes the permission cache.

Required

No.

Default

15

pkcs11.pin

This configuration parameter holds the PIN for the HSM. The IBM Security Key Lifecycle Manager obfuscates the PIN and places the obfuscated version back into the configuration file using the **`pkcs11.pin.obfuscated`** property.

pkcs11.pin

Set this parameter to hold the PIN for the HSM.

Required

Yes, if there is not already a value for the **`pkcs11.pin.obfuscated`** property.

Values

Any valid text string.

Default

None

Example

pkcs11.pin=c

pkcs11.pin.obfuscated

This configuration parameter holds the obfuscated PIN for the HSM. IBM Security Key Lifecycle Manager automatically adds this parameter.

pkcs11.pin.obfuscated

You must not use the command-line interface or any other means to enter this parameter. This parameter is derived from the **`pkcs11.pin`** parameter.

Required

Yes, if the **`pkcs11.pin`** property is not set.

Note: This property is set by IBM Security Key Lifecycle Manager.

Values

Set by IBM Security Key Lifecycle Manager. Do not modify.

Default
None

Example
pkcs11.obfuscated =
D318201607052217244A203D204B2027244D21171407214D144D

pkcs11.config

This configuration parameter holds full path and name of the HSM configuration file.

pkcs11.config
Set this parameter to hold full path and name of the HSM configuration file.

Required
Yes.

Values
Any valid file name and path.

Default
None

Example
pkcs11.pin=C:\\HSM\\Luna.cfg

requireSHA2Signatures

This property sets the algorithm of signature of certificates created by the product to SHA 256.

To modify this value, you must have permission to the configure action.

requireSHA2Signatures
Set this parameter to true to create certificates with the SHA 256 signature.

Required
Optional.

Values

true Use SHA 256 for signature creation for certificates.

false Use SHA1 for signature creation for certificates.

Default
true

Example

```
requireSHA2Signatures=true
```

rest.user.inactive_time

This property configures the inactivity time (in minutes) for users to access the IBM Security Key Lifecycle Manager REST services.

rest.user.inactive_time=timeInMinutes
The user is automatically logged out of the server if the inactive time exceeds the specified value. The user can log in again to access the IBM Security Key Lifecycle Manager REST services.

Required
Optional

Values
Integer value in minutes.

Default
15

Example
`rest.user.inactive_time=30`

stopRoundRobinKeyGrps

This property specifies whether to use keys in a key group once, or to allow additional use of keys that were previously used. 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.

stopRoundRobinKeyGrps={true|false}
Use keys in a key group once, or to allow additional use of keys that were previously used.

Required
Optional.

Values
true | false

Default
false

Example
stopRoundRobinKeyGrps=true

symmetricKeySet (replaced)

This property is removed from the `SKLMConfig.properties` file and replaced with an entry for the device group in the IBM Security Key Lifecycle Manager database. Use the **`tklmDeviceGroupAttributeList`** and **`tklmDeviceGroupAttributeUpdate`** commands to view and change the value. Previously, this property specified a key group to be used for LTO tape drives.

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

tklm.backup.db2.dir

After a backup task occurs, IBM Security Key Lifecycle Manager uses this property to record the directory that contains temporary backup files written by the **tklmBackupRun** command. The property value is specified in the **datastore.properties** file, located in the *WAS_HOME*\products\sklm\config\ directory. The parameter is not present before a backup task occurs.

tklm.backup.db2.dir=*path*
The directory that contains temporary backup files written by the **tklmBackupRun** command. If this value is not specified, the value of the **tklm.backup.dir** property is used.

Required
The parameter is not present before a backup task occurs. The database phase of the backup task must be successful before the value of this property is updated with the value of a path that the user specifies.

Values
A path that the user specifies before the backup task starts.

Default
If no value is specified, the backup task creates a *path*\temp temporary directory.

Example

```
tklm.backup.db2.dir=c:\\temp
```

tklm.backup.dir

IBM Security Key Lifecycle Manager uses this property to record the path to a directory that contains backup files written by the **tklmBackupRun** command, which specifies a user-defined path.

tklm.backup.dir=*path*
The directory that receives backup jar files written by the **tklmBackupRun** command.

Required
The backup task must be successful before IBM Security Key Lifecycle Manager updates the value of this property with a path that the user specifies. The last cleanup phase of the backup task can fail, but the property value is updated.

Values
A path that the user specifies before the backup task starts.

Default
If no value is specified, the backup operation creates a temporary backup directory under the *SKLM_HOME* directory.

Example

```
tklm.backup.dir=/var/sklm/backup
```

tklm.encryption.password

This property specifies the encryption password that IBM Security Key Lifecycle Manager uses internally to protect data in the IBM Security Key Lifecycle Manager database. This is an internally-used property. **Do not change its value.**

tklm.encryption.password=*password*
Retains the encryption password that IBM Security Key Lifecycle Manager uses internally to protect data in the IBM Security Key Lifecycle Manager database.

Required
Required.

Values
A machine-generated password string. This is an internally-used property. Do not change its value.

Default
Initially empty.

Example

```
tklm.encryption.password=5A2DD8BBA004C9F5D9C0DDD0E5EEBCBAA3D8C9
```

Note: This value is an example.

tklm.encryption.pbe.algorithm

This property specifies whether an unlimited strength algorithm is used to encrypt backup files.

tklm.encryption.pbe.algorithm=PBEWithMD5AndTripleDES
Specifies whether an unlimited strength algorithm is used to encrypt backup files written by the **tklmBackupRun** command. If you specify this property, the JVM run time must provide the required Unlimited Strength Java Cryptography Extension Policy files. Using a valid IBM ID, you can obtain the unlimited strength policy files at https://www14.software.ibm.com/webapp/iwm/web/preLogin.do?source=jcesdk

To enable the algorithm, complete these steps:

1. Stop the WebSphere Application Server.
2. Manually update the value of this property.
3. Restart the WebSphere Application Server.

Required
Required if the unlimited strength algorithm is to be used.

Values
`PBEWithMD5AndTripleDES`

Default
If no value is specified, the unlimited strength algorithm is not used.

Example

```
tklm.encryption.pbe.algorithm=PBEWithMD5AndTripleDES
```

TransportListener.ssl.ciphersuites

This property specifies supported cipher suites to be used for SSL communication between IBM Security Key Lifecycle Manager and drives.

TransportListener.ssl.ciphersuites=*cipherSuiteName*

A cipher suite describes the cryptographic algorithms and handshake protocols Transport Layer Security (TLS) and Secure Sockets Layer (SSL) use for data transfer.

Required

Optional. Required if SSL ports are specified.

Values

Values – any cipher suites that are supported by IBMJSSE2.

Default

JSSE_ALL

Examples

```
TransportListener.ssl.ciphersuites=JSSE_ALL
```

```
TransportListener.ssl.ciphersuites=TLS_RSA_WITH_RC4_128_MD5
```

Note: By specifying TLS or SSL on a cipher, you cannot determine the actual protocol (SSL or TLS) that is used by the IBM Security Key Lifecycle Manager server.

TransportListener.ssl.clientauthentication

This property specifies SSL authentication needed for communication between devices and the IBM Security Key Lifecycle Manager server.

TransportListener.ssl.clientauthentication={0|1|2}

SSL authentication needed for communication between devices and the IBM Security Key Lifecycle Manager server.

Required

Optional. Required if the SSL port value is specified.

Values

0 - No client authentication.
1 - Server can do client authentication with the client.
2 - Server must do client authentication with the client.

Default

0

Example

```
TransportListener.ssl.clientauthentication= 1
```

TransportListener.ssl.port

This property specifies the port on which IBM Security Key Lifecycle Manager server listens for requests from tape libraries that communicate using the SSL protocol.

TransportListener.ssl.port=*portnumber*

Port on which IBM Security Key Lifecycle Manager server listens for requests from tape libraries that communicate using the SSL protocol.

Required

Yes.

Values

An integer value that is a valid port number.

Default

The SSL default is 441.

Example

```
TransportListener.ssl.port=441
```

TransportListener.ssl.protocols

This property specifies security protocols.

TransportListener.ssl.protocols={**SSL_TLS**|**SSL**|**TLS**|**SSL_TLSv2**}

Specifies security protocols.

Required

Optional. Required if SSL port is specified.

Values

SSL_TLS | SSL | TLS | SSL_TLSv2

Default

SSL_TLS

Example

```
TransportListener.ssl.protocols=SSL_TLS
```

TransportListener.ssl.timeout

This property specifies how long the socket waits on a read() before closing. This property is used for the SSL socket.

TransportListener.ssl.timeout=*integerMinutes*

Specifies how long the socket waits on a read() before closing.

Required

Optional. Required if the port is specified.

Value Integer value in minutes. Do not use zero (no timeout) as a value.

Default

10

Example

```
TransportListener.ssl.timeout=10
```

TransportListener.tcp.port

This property specifies the TCP port on which IBM Security Key Lifecycle Manager server listens for requests from devices.

TransportListener.tcp.port=*portNumber*

Port on which IBM Security Key Lifecycle Manager server listens for requests from devices.

Required

Yes.

Values

An integer value that is a valid port number.

Default

3801

Example

```
TransportListener.tcp.port=3801
```

TransportListener.tcp.timeout

This property specifies how long a socket waits on a read() before closing.

TransportListener.tcp.timeout=*integerMinutes*

Specifies how long a socket waits on a read() before closing.

Required

Optional. Required if TCP port is specified.

Values

Integer value in minutes. Do not use zero (no timeout) as a value.

Default

10

Example

```
TransportListener.tcp.timeout=10
```

useSKIDefaultLabels

This property specifies whether IBM Security Key Lifecycle Manager creates externally encrypted data keys for encrypted 3592 cartridges and also DS8000 Turbo drive storage images using the X509 Subject Key Identifier (SKI) hash instead of the label when a default alias, either from the drive table or configuration file, is used.

To modify this property, you must have a role with permission to modify a 3592 device group or DS8000 device group.

useSKIDefaultLabels={**true**|**false**}

This property has no function for LTO tapes.

Required

Optional.

Values

true | false

If `true` is specified, the externally encrypted data keys are created using the SKI hash on the certificate. If `false` is specified, or the value is not specified, the default label of the certificate is used.

Default

false

Example

```
useSKIDefaultLabels=true
```

zOSCompatibility

Use this option if you plan to exchange tapes between z/OS systems running at ICSF 7740 or below, or to exchange tapes between IBM Security Key Lifecycle Manager running on a distributed system and z/OS systems running at ICSF 7740 or below, with Encryption Key Manager.

zOSCompatibility={**true**|**false**}

This option affects all keystores that might be configured to the IBM Security Key Lifecycle Manager on all platforms that are supported by the IBM Security Key Lifecycle Manager.

Required

Optional.

Values

true The encrypted tape can be read by an instance of Encryption Key Manager running on the z/OS system.

If `true` is specified, Triple Data Encryption Standard (Triple DES or DESede) symmetric keys are used. AES symmetric keys are not used. Set the value to `true` to create keys and certificates for use with z/OS systems at or below Integrated Cryptographic Services Facility version 7740.

false Default. The encrypted tape can be read by an instance of Encryption Key Manager running on the z/OS system running with ICSF 7751 or higher or with a distributed environment.

Note:

- If you specify `false`, AES symmetric keys are used. Set the value to `false` to create keys and certificates for use with z/OS systems or distributed systems when Integrated Cryptographic Services Facility version 7751 is installed.
- AES keys that you create with zOSCompatibility set to `false` (off) are not used if you later set the value of zOSCompatibility to `true` (on).

Example

```
zOSCompatibility=true
```

IBM Security Key Lifecycle Manager replication configuration parameters

You can configure parameters in the `ReplicationSKLMgrConfig.properties` file that specify various aspects of IBM Security Key Lifecycle Manager automated replication, such as server names, port numbers, file sizes, and other replication information.

Note: For the automated replication:

- The backup parameters are required only on a master system.
- The restore parameters are required only on a clone system.
- You must specify the **`replication.role`** parameter on a clone system.

replication.role

This parameter specifies whether the IBM Security Key Lifecycle Manager instance is a clone or a master.

replication.role

You can replicate only one master with a maximum of five clones.

Required

Yes. Required only on clone systems.

Values

master | clone

Default

master

replication.BackupDestDir

This parameter specifies the file system directory location to store the backup files.

replication.BackupDestDir

Set this parameter to specify the file system directory location to store the backup files.

Required

Optional.

Default

`<WAS_HOME>/products/sklm/restore`

replication.auditLogName

This parameter specifies the path and file name to store the replication audit log file.

replication.auditLogName

Set this parameter to specify the path and file name that is used as the base name to create replication audit log files, which is relative to the *SKLM_HOME* directory.

Required

Optional.

Default

`<SKLM_HOME>/logs/replication/replication_audit.log`

replication.MaxLogFileSize

This parameter specifies maximum size for the log file.

replication.MaxLogFileSize

Set this parameter to specify the log file size in kilobytes.

Required

Optional.

Values

Must be a positive integer between 100 and 500000.

Default

1000

replication.MaxLogFileNum

This parameter specifies the maximum number of log files to keep.

replication.MaxLogFileNum

Set this parameter to specify the maximum number of log files to keep.

Required

Optional.

Values

Must be a positive integer between 2 and 100.

Default

3

replication.MaxBackupNum

This parameter specifies the maximum number of backup files to keep.

replication.MaxBackupNum
Set this parameter to specify the maximum number of backup files to keep.

Required
Optional.

Values
Must be a positive integer between 2 and 10.

Default
10

replication.MasterListenPort

This parameter specifies the port number to be used for communication when unserialized or delayed replications take place.

replication.MasterListenPort
The port number must be the same on both master and clone systems.

Required
Yes.

backup.CheckFrequency

This parameter specifies the frequency to check whether backup is necessary.

backup.CheckFrequency
You must specify the frequency in minutes. This parameter is ignored if the **backup.DailyStartReplicationBackupTime** property is coded.

Required
Optional.

Values
Must be a positive integer of 1 or above

Default
60

backup.DailyStartReplicationBackupTime

This parameter specifies the replication start time in 24-clock format – HH:MM.

backup.DailyStartReplicationBackupTime
Set this parameter to specify the replication start time in 24-clock format – HH:MM.

Required
Optional.

backup.SerializeRestores

This parameter specifies whether to serialize the first restore. The backups are always serialized if replication is initiated from the CLI.

backup.SerializeRestores
If the value of this parameter is true, wait for a message from the first clone system, signaling a successful restore, before continuing to send backup files to other clones.

Required
Optional.

Values
true | false

Default
false

backup.ClientIP1

This parameter specifies the IP address or hostname for clone 1.

backup.ClientIP1
Set this parameter to specify the IP address or hostname for clone 1.

Required
Yes. Specify this parameter minimally.

backup.ClientIP2

This parameter specifies the IP address or hostname for clone 2.

backup.ClientIP2
Set this parameter to specify the IP address or hostname for clone 2.

Required
Optional.

backup.ClientIP3

This parameter specifies the IP address or hostname for clone 3.

backup.ClientIP3
Set this parameter to specify the IP address or hostname for clone 3.

Required
Optional.

backup.ClientIP4

This parameter specifies the IP address or hostname for clone 4.

backup.ClientIP4
Set this parameter to specify the IP address or hostname for clone 4.

Required
Optional.

backup.ClientIP5

This parameter specifies the IP address or hostname for clone 5.

backup.ClientIP5
Set this parameter to specify the IP address or hostname for clone 5.

Required
Optional.

backup.ClientPort1

This parameter specifies the port number to use for sending files to clone 1.

backup.ClientPort1
Set this parameter to specify the port number to use for sending files to clone 1.

Required
Yes. Specify this parameter minimally.

backup.ClientPort2

This parameter specifies the port number to use for sending files to clone 2.

backup.ClientPort2
Set this parameter to specify the port number to use for sending files to clone 2.

Required
Optional.

backup.ClientPort3

This parameter specifies the port number to use for sending files to clone 3.

backup.ClientPort3
Set this parameter to specify the port number to use for sending files to clone 3.

Required
Optional.

backup.ClientPort4

This parameter specifies the port number to use for sending files to clone 4.

backup.ClientPort4
Set this parameter to specify the port number to use for sending files to clone 4.

Required
Optional.

backup.ClientPort5

This parameter specifies the port number to use for sending files to clone 5.

backup.ClientPort5
Set this parameter to specify the port number to use for sending files to clone 5.

Required
Optional.

backup.EncryptionPassword

This parameter specifies the encryption password for backups.

backup.EncryptionPassword
Set this parameter to specify the encryption password for backups, and when read in, obfuscates and changes to `backup.ObfuscatedEncryptionPassword`.

Required
Yes. You must specify either this parameter or the `backup.ObfuscatedEncryptionPassword` parameter.

backup.ObfuscatedEncryptionPassword

This parameter specifies the password for backups obfuscated.

backup.ObfuscatedEncryptionPassword

IBM Security Key Lifecycle Manager generates this parameter and you must not modify it.

Required

Yes. You must specify either this parameter or the `backup.EncryptionPassword` parameter.

backup.BackupDescriptionText

This parameter specifies freeform text descriptor for the backup.

backup.BackupDescriptionText

Set this parameter to specify freeform text descriptor for the backup. The description is limited to a maximum of 100 characters.

Required

Optional.

backup.TLSCertAlias

This parameter specifies alias of the certificate that is used for the TLS/SSL client.

backup.TLSCertAlias

Set this parameter to specify the alias of the certificate that is used for the TLS/SSL client.

Required

Yes.

backup.ReleaseKeys

This parameter specifies whether IBM Security Key Lifecycle Manager must release the keys for use after a successful backup or restore.

backup.ReleaseKeys

This parameter is ignored if you do not specify the enableKeyRelease IBM Security Key Lifecycle Manager configuration parameter.

Required

Optional.

Values

OFF | BACKUP | RESTORE

Default

OFF

restore.ListenPort

This parameter specifies the port number that the clone system must listen on to receive backups.

restore.ListenPort

Set this parameter to specify the port number that the clone system must listen on to receive backups.

Required

Yes. Required only on clone systems.

restore.DailyStartReplicationRestoreTime

This parameter specifies the time in 24-clock HH:MM format to restore backup.

restore.DailyStartReplicationRestoreTime

Set this parameter to specify the time in 24-clock HH:MM format to restore backup. If you do not specify the time, restore is done as soon as the backup is available.

Required

Optional.

restore.TipadminUsername

This parameter specifies the user ID of TIP admin.

Note: This **`restore.TipadminUsername`** parameter is deprecated.

restore.TipadminUsername

Set this parameter to specify the user ID of TIP admin. You require this parameter only on Linux platform to restart IBM Security Key Lifecycle Manager after a completed replication.

Required

Yes. Only for Linux platform.

restore.TipadminPassword

This parameter specifies the password for TIP admin.

Note: This **`restore.TipadminPassword`** parameter is deprecated.

restore.TipadminPassword

Set this parameter to specify the password for TIP admin. You require this parameter only on Linux platform to restart IBM Security Key Lifecycle Manager after a completed replication. IBM Security Key Lifecycle Manager obfuscates this parameter in the configuration file.

Required

Yes. Required only for Linux platform. You must specify either this parameter or the **`restore.ObfuscatedTipadminPassword`** parameter.

restore.ObfuscatedTipadminPassword

This parameter specifies the obfuscated TIP admin password.

Note: This **`restore.ObfuscatedTipadminPassword`** parameter is deprecated.

restore.ObfuscatedTipadminPassword

Set this parameter to specify the obfuscated TIP admin password. You require this parameter only on Linux platform to restart IBM Security Key Lifecycle Manager after a completed replication. The IBM Security Key Lifecycle Manager generates this password and you must not set it.

Required

Yes. Required only for Linux platform. You must specify either this parameter or the **`restore.TipadminPassword`** parameter.

Changes to configuration properties or database values

Changes to some configuration properties in the SKLMConfig.properties file or in the IBM Security Key Lifecycle Manager database can occur dynamically.

Changes to other properties or database entries require that you restart the IBM Security Key Lifecycle Manager server before the change takes effect.

Depending on the change you intend to make, you might use the graphical user interface or the command-line interface. Not all properties in the SKLMConfig.properties file or in the IBM Security Key Lifecycle Manager database can be changed with both interfaces.

Table 4. Changes to configuration properties or database entries

Property	Installation sets default	Changes occur dynamically	Change requires server restart	Change available only in command-line interface
Audit.event.outcome	✓		✓	
Audit.eventQueue.max	✓		✓	
Audit.event.types	✓	✓		
Audit.handler.file.multithreads			✓	✓
Audit.handler.file.name	✓		✓	
Audit.handler.file.size	✓		✓	
Audit.handler.file.threadlifespan			✓	✓
backup.keycert.before.serving	✓	✓		✓
cert.valiDATE		✓		
config.keystore.name		You can change this property only once by using the graphical user interface.	✓	You cannot modify this property by using the command-line interface.
config.keystore.ssl.certalias *		✓	✓	
device.AutoPendingAutoDiscovery (an attribute in the IBM Security Key Lifecycle Manager database)		✓		
device.enableMachineAffinity (an attribute in the IBM Security Key Lifecycle Manager database)	✓	✓		
drive.acceptUnknownDrives (replaced by device group attribute **device.AutoPendingAutoDiscovery** in the IBM Security Key Lifecycle Manager database)				
drive.default.alias1 (replaced by a device group attribute in the IBM Security Key Lifecycle Manager database)				
drive.default.alias2 (replaced by a device group attribute in the IBM Security Key Lifecycle Manager database)				

Table 4. Changes to configuration properties or database entries (continued)

Property	Installation sets default	Changes occur dynamically	Change requires server restart	Change available only in command-line interface
ds8k.acceptUnknownDrives (replaced by device group attribute device.AutoPendingAutoDiscovery in the IBM Security Key Lifecycle Manager database)				
enableClientCertPush		✓	✓ (if changed manually)	
fips				✓
KMIPListener.ssl.port *		✓	✓	
lock.timeout		✓		✓
maxPendingClientCerts		✓	✓ (if changed manually)	
pcache.refresh.interval	This property is optional in the configuration file. By default, its value is not set and IBM Security Key Lifecycle Manager uses the default time interval of 15 minutes.	✓		✓
symmetricKeySet (an attribute in the IBM Security Key Lifecycle Manager database)				
tklm.backup.db2.dir				You cannot modify this property by using the command-line interface.
tklm.backup.dir	Running a backup adds this property to the configuration file.		✓	You cannot modify this property by using the command-line interface.
tklm.encryption.password	This is an internally used property. Do not change its value. You cannot modify this property by using the command-line interface.			

Table 4. Changes to configuration properties or database entries (continued)

Property	Installation sets default	Changes occur dynamically	Change requires server restart	Change available only in command-line interface
tklm.encryption.pbe.algorithm	✓		✓	
TransportListener.tcp.port	✓		✓	
TransportListener.tcp.timeout			✓	
TransportListener.ssl.ciphersuites			✓	✓
TransportListener.ssl.clientauthentication				✓
TransportListener.ssl.port *		✓	✓	
TransportListener.ssl.protocols			✓	
TransportListener.ssl.timeout			✓	
stopRoundRobinKeyGrps				✓
useSKIDefaultLabels		✓		
zOSCompatibility		✓		
* If you set this value for the first time, restart is not required. If you later modify the value, restart is required.				

Audit records on distributed systems

When auditable events occur on distributed systems during request processing, the IBM Security Key Lifecycle Manager audit subsystem writes textual audit records to a set of sequential files. You can also generate the audit records in syslog format and send them to a syslog server.

You can configure the IBM Security Key Lifecycle Manager configuration properties file to generate audit records in syslog format and send them to a syslog server. To send the audit records to the server, you must specify the server host name or IP address, and the port number in the properties file.

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 server host name and the port number are not specified.

If the host name or IP address of the syslog server is not reachable, audit log messages are redirected to a local audit file in syslog format. When the server is up, the logs are directed to the server.

If you do not enable the syslog format for the audit records, the textual audit records are written to a set of sequential files.

When a current file reaches the configurable size, the audit subsystem closes and renames the file with a time stamp, and opens the next file to which audit records are written. The overall audit log is the set of sequentially named files.

To limit the total number of audit record files, you might create a script or program to monitor the set of files in the audit directory. As files are closed and named based on the timestamp, your script might copy and append the file contents to a permanent log file and directory that you specify, and then delete the file. Ensure that you do not remove or alter the active file to which IBM Security Key Lifecycle Manager is writing records.

Note: Audit record formats are not considered to be programming interfaces. The format of these records might change from release to release.

Audit configuration properties

You can configure properties that specify the audit record directory, file name, file size, event types, and other auditing activities.

Edit the `SKLMConfig.properties` configuration file to configure auditing properties.

Audit.event.outcome

Specifies whether events are occurring as a result of successful operations, unsuccessful operations, or both are audited.

Note: Do not delete this property from the properties file.

Audit.event.outcome={*outcome*[,*outcome*]}

Specify **success** for events to be logged which occur as a result of successful operations. Specify `failure` for events to be logged which occur as a result of unsuccessful operations. Only audit events that resulted in the specified outcome are recorded.

Required

Yes.

Values

success | failure

Both can be specified separated by comma or semicolon.

Default

success, failure

Examples

Specification for this configuration value is shown in the following example.

```
Audit.event.outcome=failure
```

To enable both successful and unsuccessful cases:

```
Audit.event.outcome=success,failure
```

Audit.eventQueue.max

This property sets the maximum number of event objects to be held in the memory queue before they are flushed to file.

Audit.eventQueue.max=*numberOfEvents*

This property is optional but suggested. The default is zero.

Required

Optional. Suggested.

Values

0 - *numberOfEvents*

A value of zero means flush immediately. *numberOfEvents* is an integer greater than zero.)

Default

0

Note: A default value of zero causes all events to be written to a file.

Example

```
Audit.eventQueue.max=0
```

To avoid first failure data loss, *do not change* the queue maximum to a value other than zero.

Audit.event.types

This property specifies which audit types are sent to the audit log.

Audit.event.types={*type*[,*type*]}

Only audit events that resulted in the specified outcome are recorded.

Required

Yes.

Values

all | runtime| authentication | authentication_terminate | authorization | configuration_management | resource_management | key_management | none

Multiple values can be specified, separated by a comma or semicolon.

Note: Do not specify a value of `none` in combination with other values.

all All event types.

authentication

Authentication events.

authentication_terminate

Events that occur when the user logs out. Not used.

authorization

Authorization events.

configuration_management

Configuration events.

key_management

Events when changes occur in the configuration of keys.

none No events.

Note: Do not specify a value of none in combination with other values.

resource_management

Events when changes occur in the configuration of resources such as tape drives to the IBM Security Key Lifecycle Manager server.

runtime

Events that occur as a part of processing operations and requests that are sent to IBM Security Key Lifecycle Manager.

Default

authorization, authorization_terminate, resource_management, runtime, key_management

Examples

To collect all auditable event data, enter:

```
Audit.event.types=all
```

Another example is:

```
Audit.event.types=authorization,runtime,resource_management
```

Audit.handler.file.name

This property specifies the path and file name to which audit entries are logged, such as sklm_audit.log.

Audit.handler.file.name=*path***/***fileName*

This is the path and file name that is used as the base name in creating audit log files in the specified audit directory, which is relative to the *SKLM_HOME* directory.

Required

Yes.

Default

logs/audit/sklm_audit.log

Example

To set the base name to my_sklm_audit.log, enter:

```
Audit.handler.file.name=my_sklm_audit.log
```

The name of the audit log file that reaches its maximum size appends a value for the time at which the file was closed. For example, if **Audit.handler.file.name** value is set to sklm_audit.log, a closed file has a name like sklm_audit.log.2315003554. Higher number values indicate newer audit log files.

Audit.handler.file.size

This property specifies the size in KB that the Audit.handler.file.name file grows before a new file is created.

Audit.handler.file.size=*sizeInKBytes*

The value indicates the size limit at which an audit file is closed and a new audit file is written. The actual size of the resulting audit file might exceed this value by several bytes because the file is closed after the size limit is exceeded.

Required
Optional. Recommended.

Values
0 - *integervalue* (in KB).

The value of *integervalue* is a positive integer. Specifying zero sets the file size to the default value.

Default
10000

The default is 10 MB, which is 10000 KB.

Example
To set the file size to approximately 20 MB, enter:

```
Audit.handler.file.size=20000
```

Audit.handler.file.threadlifespan

This property limits the lifetime of an audit record processing thread if the value of **audit.handler.file.multithreads** is set to true.

Audit.handler.file.threadlifespan=*timeInSeconds*
This value is used during cleanup processing to allow threads to complete their work before the threads are interrupted. If a background thread is not completed its work within the life span, cleanup processing interrupts the thread.

Required
Optional.

Values
Specified in seconds.

Default
10

Example
To set the expected time to 15 seconds as the interval of time for a thread to write to the audit log, enter:

```
Audit.handler.file.threadlifespan=15
```

Audit.handler.file.multithreads

This property specifies whether the audit handler dispatches separate threads to process audit records.

Audit.handler.file.multithreads={**true**|**false**}
Specifies whether to use separate threads to process audit records.

Required
Optional.

Values
true | false

If the property is set to true, a separate thread is used to write the event data to the audit log, allowing the current thread of execution (operation) to continue without waiting for the write to the audit log to complete. Use of multiple threads is the default behavior.

Default
true

Example

To set multithreading to `false`, enter:

```
Audit.handler.file.multithreads=false
```

Audit.isSyslog

The `Audit.isSyslog` property specifies whether the audit log messages are generated in syslog format in a syslog server. To redirect audit messages to the syslog server, you must also specify the host name and port number of the server.

Audit.isSyslog={true|false}

If the value of this property is `false`, the audit log messages are written to a configured local audit file.

Required

Optional.

Values

`true | false`

Default

`false`

Example

To enable the syslog format, enter:

```
Audit.isSyslog=true
```

Audit.syslog.server.host

The `Audit.syslog.server.host` property specifies the host name or IP address of the syslog server where the IBM Security Key Lifecycle Manager audit log messages are sent.

Audit.syslog.server.host=*host name***/***IP address*

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 server host name and the port number are not specified.

Required

Optional.

Values

The value cannot exceed 256 characters in length.

If the host name or IP address of the syslog server is not reachable, audit log messages are redirected to a local audit file in syslog format.

Example

To set IP address of the syslog server, enter:

```
Audit.syslog.server.host=9.118.42.23
```

To redirect audit log messages to the syslog server, you must also specify the value for `Audit.syslog.server.port`.

Audit.syslog.server.port

The `Audit.syslog.server.port` property specifies the port number on which the syslog server listens for requests.

Audit.syslog.server.port=*port number*

The audit log messages are written to a configured local audit file in syslog format for the following reasons:

- Syslog format is enabled for the audit messages.
- Syslog server host name and the port number are not specified.

Required
Optional.

Values
An integer value 1 - 65535.

Example
To specify port number for the syslog server, enter:

```
Audit.syslog.server.port=1000
```

To redirect audit log messages to the syslog server, you must also specify the value for `Audit.syslog.server.host`.

Audit.syslog.isSSL

The `Audit.syslog.isSSL` property specifies whether the SSL/TLS transport protocol is enabled for secure transfer of audit log messages to the syslog server.

Audit.syslog.isSSL={**true**|**false**}

Required
Optional.

Values
`true | false`

Default
`false`

Example
To use the SSL/TLS transport protocol for secure message transmission, enter:

```
Audit.syslog.isSSL=true
```

Audit record format

All audit records use a similar output format.

All audit records contain some common information including time stamp and record type, along with information specific to the audit event that occurred. Installing or starting IBM Security Key Lifecycle Manager writes the build level to the audit log.

The general format for audit records is:

```
AuditRecordType:[
  timestamp=timestamp
  Attribute Name=Attribute Value
  ...
  ]
```

Each record spans multiple lines in the file, with the first line of the record beginning with the audit record type beginning at the first character on the line, followed by a colon (;) and an opening left bracket ([).

Subsequent lines associated with the same audit record are indented two (2) spaces to assist in readability of the log records. The last line for a single audit record contains a closing right bracket (]) indented two (2) spaces. The number of lines for each audit record varies based on the audit record type and the additional attribute information that is provided with the audit record.

The timestamp for the audit records is based on the system clock of the system on which IBM Security Key Lifecycle Manager is running. If these records are to be correlated based on timestamp with events occurring on other systems, use some type of time synchronization to ensure that the clocks of the various systems in the environment are synchronized to an acceptable level of accuracy.

Audited events

Auditable events cause IBM Security Key Lifecycle Manager to create audit records.

This table lists the audit record type that is logged when an event occurs:

Table 5. Audit record types by audited event

Audited Event	Audit Record Type
User successfully authenticated	authentication
User authentication failed	authentication
Authorization successful	authorization
Authorization failure	authorization
Command-line processing started	runtime
exit command received	runtime
Unknown command entered	runtime
Message received from drive	runtime
Error processing message from drive	runtime
Error from message received from drive	runtime
Error updating drive table with information received from drive	runtime
Error retrieving information from drive table	runtime
Error retrieving information from keystore	runtime
Error processing certificate from keystore	runtime
Error finding private key from keystore	runtime
Error computing cryptographic values	runtime
Message exchange processed successfully	runtime
Message processing started	runtime
Command-line processing started	runtime
Problem found using cryptographic services	runtime
New drive discovered	runtime
Error configuring drive to drive table	runtime
Successfully started processing messages from drive	runtime
Drive removed from drive table	resource_management
Error removing drive from drive table	resource_management
Drive table import successful	resource_management
Error importing drive table	resource_management

Table 5. Audit record types by audited event (continued)

Audited Event	Audit Record Type
Drive table export successful	resource_management
Error exporting drive table	resource_management
Drive add to drive table successful	resource_management
Error adding drive to drive table	resource_management
Drive table modify successful	resource_management
Error modifying drive table	resource_management
Successful keystore open	resource_management
Error opening keystore	resource_management
Configuration property changed	configuration_management
Error changing configuration property	configuration_management
Configuration property deleted	configuration_management
Error deleting configuration property	configuration_management
Configuration import successful	configuration_management
Error importing configuration	configuration_management
Configuration export successful	configuration_management
Error exporting configuration	configuration_management
REST service user logoff due to inactivity	authentication
REST user attempting to access REST service without authentication ID	authentication
Unauthenticated or logged out REST user attempting to access REST service	authentication
REST service user initiated logout	authentication
REST service user successfully authenticated	authentication
REST service user authentication failed	authentication

Messages

Depending on the outcome of an operation, IBM Security Key Lifecycle Manager might provide an informational, warning, or error message.

Message syntax

The message syntax contains elements for the product identifier, as well as which part of the product issued the message, the message number, and an indicator that the message content contains information, a warning, or error description.

Messages have the following syntax:

```
CTGUUXXXXZ
```

where:

CTG Identifies the IBM Security Key Lifecycle Manager product.

UU Identifies which part of the product issued the message. For example:

- **KM** The IBM Security Key Lifecycle Manager server issued the message.

KO Password policy messages.

KS The IBM Security Key Lifecycle Manager key server issued the message.

XXXX Is the message number, such as 0001.

Z Is the character I for informational message, W for warning message, or E for error message.

For example:

```
CTGKM0545E: An error occurred exporting a certificate.
```

Error and warning messages

These are the IBM Security Key Lifecycle Manager error and warning messages.

File Deleted Antigen for Exchange removed getadmingroupname.vbs since it was found to match the FILE FILTER= unnamed: *.vbs file filter.

Explanation: The problem is typically caused on Windows systems by antivirus installation software.

System action: Installation fails.

Administrator response: Take these steps:

1. Obtain and reinstall the `getadmingroupname.vbs` file.
2. Change the antivirus file filter to avoid removing this file.
3. Install IBM Security Key Lifecycle Manager again.

CTGKM0002E Command failed: *VALUE_0*

Explanation: The specification of the command, or one or more parameters in the command, is incorrect.

System action: The command fails.

Administrator response: Examine the error message, which might indicate which parameter caused the error. Retype the command string, and then try the command again.

CTGKM0003E Unhandled exception.

Explanation: Unhandled exception.

System action: The command fails.

Administrator response: Examine the exception message, and then try the command again.

CTGKM0100E Cannot obtain audit and key serving parameters information.

Explanation: An internal component of the IBM Security Key Lifecycle Manager server is not running, such as the key server.

System action: The internal component fails to obtain audit and debug information.

Administrator response: Contact IBM Support.

CTGKM0101E Cannot update audit information.

Explanation: A change was not written to the SKLMConfig.properties file.

System action: A property value is not updated.

Administrator response: Ensure that the SKLMConfig.properties file is write enabled. Then, try the operation again. If the problem persists, contact IBM Support.

CTGKM0102E Cannot update key serving parameter or port information.

Explanation: A change was not written to the SKLMConfig.properties file.

System action: A property value is not updated.

Administrator response: Ensure that the SKLMConfig.properties file is write enabled. Then, try the operation again. If the problem persists, contact IBM Support.

CTGKM0103E Cannot retrieve keystore information.

Explanation: The keystore information was not available from the IBM Security Key Lifecycle Manager database.

System action: The database is not available.

Administrator response: Ensure that IBM Security Key Lifecycle Manager database is available. Correct any database server runtime errors that you identify. Then, try the operation again.

CTGKM0104E Cannot add keystore.

Explanation: The keystore information might not be available from the IBM Security Key Lifecycle Manager database. Alternatively, the directory for the keystore file cannot be found. You might not have access to the directory. There might be more information in the message that describes the problem.

System action: The keystore is not added.

Administrator response: Ensure that the IBM Security Key Lifecycle Manager database is available. Additionally, determine that the directory exists for the keystore file. Additional information in the message might guide your response. Make appropriate changes. Then, try the operation again.

CTGKM0110E Retrieval of SSL/KMIP certificates failed:

Explanation: The attempt to obtain a list of SSL/KMIP certificates was not successful. IBM Security Key Lifecycle Manager database might not be available, or a connection to the IBM Security Key Lifecycle Manager server might not be available.

System action: A list of certificates is not retrieved.

Administrator response: Ensure that connections to the IBM Security Key Lifecycle Manager database and IBM Security Key Lifecycle Manager server are available. If a connection problem exists, make the appropriate corrections. Then, try the operation again.

CTGKM0111E Retrieval of IKEv2-SCSI certificates failed:

Explanation: The attempt to obtain a list of IKEv2-SCSI certificates was not successful. IBM Security Key Lifecycle Manager database might not be available, or a connection to the IBM Security Key Lifecycle Manager server might not be available.

System action: A list of certificates is not retrieved.

Administrator response: Ensure that connections to IBM Security Key Lifecycle Manager database and IBM Security Key Lifecycle Manager server are available. If a connection problem exists, make the appropriate corrections. Then, try the operation again.

CTGKM0112E SSL/KMIP certificate submit failed:

Explanation: You might not have permission to write to the certificate request file. Alternatively, there might not be sufficient free disk space, or the database might not be available.

System action: The certificate request is not created.

Administrator response: Ensure that your permissions are correct, that there is sufficient free disk space, and that the database connection is available. If not, make the appropriate corrections. Then, try the operation again.

CTGKM0113E IKEv2-SCSI certificate submit failed:

Explanation: You might not have permission to write to the certificate request file. Alternatively, there might not be sufficient free disk space, or the database might not be available.

System action: The certificate request is not created.

Administrator response: Ensure that your permissions are correct, that there is sufficient free disk space, and that the database connection is available. If not, make the appropriate corrections. Then, try the operation again.

CTGKM0114E Certificate label and description (common name) are required fields.

Explanation: An empty or null value was found for a certificate name, or the description (common name) for the certificate might be missing in the certificate request.

System action: The operation fails.

Administrator response: Specify a unique value for the certificate name and specify a common name for the certificate. Then, try the operation again.

CTGKM0115E Certificate was not selected.

Explanation: A certificate was not selected from the list of valid, active certificates for communication with the server.

System action: The operation fails.

Administrator response: Select a valid certificate, Then, try the operation again.

CTGKM0116E Selected certificate does not match any active certificates in the keystore.

Explanation: An exception occurred in internal processes.

System action: The selected certificate does not match any of the list of certificates in the keystore. The certificate might have been manually deleted from the keystore, but not from metadata in the IBM Security Key Lifecycle Manager database.

Administrator response: Select a different certificate. Then, try the operation again. You might need to ensure that your database metadata is a match to the contents of the keystore.

CTGKM0117E Cannot create directory *DIRECTORY_NAME* .

Explanation: You might not have the correct access to create the directory, or the specified path might have an error. The directory might already exist. There might be additional information.

System action: The create directory operation fails.

Administrator response: Ensure that your access allows you to create a directory and that the specified path is valid. Additional information might also guide your response. Make appropriate changes. Then try the operation again.

CTGKM0118E Directory with the name you specified already exists.

Explanation: The specified directory name already exists in the file system.

System action: The create directory operation fails.

Administrator response: Ensure that the specified directory does not already exist in the file system.

CTGKM0119E Cryptographic object not found: *VALUE_0*

Explanation: The cryptographic object for given unique identifier or name does not found.

System action: The PUSH operation fails.

Administrator response: Ensure that you have provided correct unique identifier or name for the cryptographic object to be pushed to client.

CTGKM0120E Either cryptographic object's unique identifier(uuid) or name is required. Please provide one.

Explanation: Either cryptographic object's unique identifier(uuid) or name is required. Please provide one.

System action: The PUSH operation fails.

Administrator response: Ensure that either cryptographic object's unique identifier(uuid) or name is provided for PUSH operation.

CTGKM0200E Cannot add device.

Explanation: The device might already exist, or the IBM Security Key Lifecycle Manager database might not be available. There might be additional information.

System action: The device add operation fails.

Administrator response: Determine whether you correctly specified the device. Alternatively, you might need to confirm that the database is available. Additional information in this message might also guide your response. Make corrections. Then, try the operation again.

CTGKM0201E Cannot modify device.

Explanation: The device might already exist, or the IBM Security Key Lifecycle Manager database might not be available. There might be additional information.

System action: The device modify operation fails.

Administrator response: Determine whether you correctly specified the device. Alternatively, you might need to confirm that the database is available. Additional information in this message might also guide your response. Make corrections. Then, try the operation again.

CTGKM0202E Cannot delete device.

Explanation: The device might not exist, or the IBM Security Key Lifecycle Manager database might not be available. There might be additional information.

System action: The device delete operation fails.

Administrator response: Determine whether you correctly specified the device. Alternatively, you might need to confirm that the database is available. Additional information in this message might also guide your response. Make corrections. Then, try the operation again.

CTGKM0205E Cannot retrieve template defaults.

Explanation: Internally-processed template default information was not available from the IBM Security Key Lifecycle Manager database.

System action: The operation fails.

Administrator response: Determine whether the IBM Security Key Lifecycle Manager database is available. Correct any database server runtime errors that you identify. Then, try the operation again. You might need to contact IBM Support.

CTGKM0206E Cannot retrieve certificates.

Explanation: A list of certificates was not available. The IBM Security Key Lifecycle Manager database might not be available. There might be additional information.

System action: Certificate retrieval fails.

Administrator response: Confirm that the database is available. You might need to restart the IBM Security Key Lifecycle Manager server. Additional information in this message might also guide your response. Make corrections. Then, try the operation again.

CTGKM0207E Cannot create certificate.

Explanation: IBM Security Key Lifecycle Manager could not create a certificate in the keystore. There might be a problem with the certificate information or a problem with the keystore. There might be additional information.

System action: The certificate create operation fails.

Administrator response: Ensure that the required certificate information is correct. You might use the tklmKeyStoreList command to ensure that the keystore is available. Additional information in this message might also guide your response. Correct errors. Then, try the operation again.

CTGKM0208E Cannot modify certificate.

Explanation: IBM Security Key Lifecycle Manager could not modify information for the specified certificate. If you specify that the certificate is a system default or partner certificate, the properties file might not be available. If you modify the Trust setting for a certificate, refer to additional information in this message.

System action: The certificate modify operation fails.

Administrator response: Ensure that the properties file is available, and that you have write access. Additional information in this message might guide your response. Correct errors. Then, try the operation again.

CTGKM0209E Cannot delete certificate.

Explanation: IBM Security Key Lifecycle Manager could not delete a certificate that is a system default or partner, or associated with a device. This message might have additional information.

System action: The certificate delete operation fails.

Administrator response: Ensure that the certificate is not a system default or partner certificate, and that the certificate has no associated devices. Additional information in this message might also guide your response. Then, try the operation again. If the delete operation is successful, ensure that you back up the keystore again to retain its current state.

CTGKM0210E Cannot update setting to accept requests from all drives.

Explanation: An attempt was made to change the drive.acceptUnknownDrives property in the SKLMConfig.properties file. The file might be write protected. Alternatively, an internal component such as the key server component returned an error or was not available. This message might provide additional information.

System action: The operation fails.

Administrator response: Ensure that the SKLMConfig.properties file is write enabled. If an internal component failed, you might restart the IBM Security Key Lifecycle Manager server. Then, try the operation again. Additional information in this message might guide your response.

CTGKM0211E Cannot retrieve IBM Security Key Lifecycle Manager status.

Explanation: An internal component such as the key server component returned an error or was not available. The SKLMConfig.properties file might be write protected. This message might provide additional information.

System action: Server status retrieval fails.

Administrator response: Ensure that the SKLMConfig.properties file is write enabled. You might restart the IBM Security Key Lifecycle Manager server. Then, try the operation again. Additional information in this message might guide your response.

CTGKM0214E Cannot retrieve key groups.

Explanation: IBM Security Key Lifecycle Manager database might not be available. Alternatively, an internal component such as the key server component returned an error or was not available. This message might provide additional information.

System action: The operation fails.

Administrator response: Confirm that the database is available. You might need to restart the IBM Security Key Lifecycle Manager server. Additional information in this message might also guide your response. Make corrections. Then, try the operation again.

CTGKM0215E Cannot create key group.

Explanation: You might have specified a value that is not valid for a key group. Alternatively, IBM Security Key Lifecycle Manager database might not be available or an internal component such as the key server component returned an error or was not available.

System action: The operation fails.

Administrator response: Ensure that your values for a key group are valid. Ensure that IBM Security Key Lifecycle Manager database is available. If an internal component failed, you might restart the IBM Security Key Lifecycle Manager server. Then, try the operation again.

CTGKM0216E Cannot modify this key group.

Explanation: An attempt failed to modify this key group. There might be more information in the message that describes the problem.

System action: The modification requested for this key group does not occur. No keys are added to or deleted from the key group.

Administrator response: Additional information might guide your response. Make appropriate changes. Then try the operation again.

CTGKM0217E Cannot delete key group that is a system default, or that is associated with a device or pending device.

Explanation: IBM Security Key Lifecycle Manager could not delete a key group that is a system default, or that is associated with a device. There might be additional information in this message.

System action: The key group is retained in the IBM Security Key Lifecycle Manager database.

Administrator response: Ensure that the key group is not the system default, and that the key group also has no associated devices. Additional information in this message might also guide your response. Then, try the operation again.

CTGKM0218E Cannot create the following keys in this key group.

Explanation: An attempt failed to create keys for this key group. There might be more information in the message that describes the problem.

System action: The keys are not created, or are not added as members of the key group.

Administrator response: Additional information might guide your response. Make appropriate changes. Then try the operation again.

CTGKM0219E Because this certificate is currently expired and the validate certificates check is enabled, cannot make this certificate the System Default or System Partner.

Explanation: The certificate has expired and is no longer available for use.

System action: The operation fails.

Administrator response: Select another certificate. This certificate has expired.

CTGKM0220E Cannot retrieve available keys.

Explanation: An attempt failed to retrieve all available symmetric keys. There might be more information in the message that describes the problem.

System action: The keys are not found.

Administrator response: Additional information might guide your response. Make appropriate changes. Then try the operation again.

CTGKM0221E Cannot retrieve keys from key group.

Explanation: An attempt failed to retrieve keys from a key group. There might be more information in the message that describes the problem.

System action: The keys are not found.

Administrator response: Additional information might guide your response. Make appropriate changes. Then try the operation again.

CTGKM0222E Device serial number is not valid *VALUE_0* **, must be 12 characters for 3592 and DS8000 device families or 1-48 characters long for DS5000 device family, contain valid characters, and cannot have leading or trailing whitespace.**

Explanation: The device serial number for a device be exactly 12 characters for 3592 and DS8000 device families or 1-48 characters for DS5000 family and follow a specific format.

System action: The operation fails.

Administrator response: Ensure that your specification is 12 valid characters in length and contains alphanumeric, period, dash, semicolon, and underscore characters, or a space that is not in the first or last position. Additional information in this message might also guide your response. Correct the problem and try the operation again.

CTGKM0223E Device already exists with device serial number *VALUE_0* **, Device group** *VALUE_1* **, World Wide Name** *VALUE_2*

Explanation: You specified values for a device that already exists in the IBM Security Key Lifecycle Manager database.

System action: The device create operation fails.

Administrator response: Specify values for a device that does not currently exist in the IBM Security Key Lifecycle Manager database. Then, try the operation again.

CTGKM0224E World wide name is not valid *VALUE_0* **, must be 8 characters long or less.**

Explanation: IBM Security Key Lifecycle Manager requires that a worldwide name be 8 characters or less in length.

System action: The operation fails.

Administrator response: Specify a worldwide name that meets the length requirement of 8 characters or less. Then, try the operation again.

CTGKM0225E Cannot add device with device serial number *VALUE_0* because the specified key *VALUE_1* does not exist in keystore *VALUE_2*

Explanation: You attempted to create a device and associate it with a key that was not found in the IBM Security Key Lifecycle Manager keystore.

System action: The device add operation fails.

Administrator response: Specify an alternate key. Then, try the operation again.

CTGKM0226E The certificate is not active and cannot be the System Default or System partner certificate.

Explanation: The validate certificates check is enabled. The process determined that this certificate is not active and cannot be the System Default or System Partner.

System action: The operation fails.

Administrator response: Select a certificate that is in active state and try the operation again.

CTGKM0227E Cannot retrieve available key groups.

Explanation: The key groups information was not available from the IBM Security Key Lifecycle Manager database.

System action: The key groups are not retrieved.

Administrator response: Ensure that the IBM Security Key Lifecycle Manager database is available. Then, try the operation again.

CTGKM0228E Cannot retrieve key information.

Explanation: An attempt failed to retrieve key information. There might be more information in the message that describes the problem.

System action: The operation fails.

Administrator response: Additional information might guide your response. Make appropriate changes. Then try the operation again.

CTGKM0229E Cannot retrieve keys.

Explanation: An error occurred while retrieving a list of keys. There might be more information in the message that describes the problem.

System action: The operation fails.

Administrator response: Additional information might guide your response. Make appropriate changes. Then, try the operation again.

CTGKM0230E Cannot create keys.

Explanation: An attempt to create a key or keys did not complete. There might be more information in the message that describes the problem.

System action: The operation fails.

Administrator response: Additional information might guide your response. Make appropriate changes. Then, try the operation again.

CTGKM0231E Cannot modify key membership.

Explanation: An attempt to modify key membership did not complete. There might be more information in the message that describes the problem.

System action: The operation fails.

Administrator response: Additional information might guide your response. Make appropriate changes. Then, try the operation again.

CTGKM0232E Cannot delete key.

Explanation: The key that you intend to delete, might not be found, or there might be a database error. There might be more information in the message that describes the problem.

System action: The key is not deleted.

Administrator response: Additional information might guide your response. Make appropriate changes. Then, try the operation again.

CTGKM0233E Cannot list backup files.

Explanation: Cannot retrieve the data for the backup files.

System action: The list backup files operation fails.

Administrator response: Check the log entries to find out the reason for the failure. Make appropriate changes. Then, try the operation again.

CTGKM0234E Cannot locate default backup directory.

Explanation: Cannot read the property value for the backup directory.

System action: Failed to read backup directory.

Administrator response: Check the log entries to find out the reason for the failure. Make appropriate changes. Then, try the operation again.

CTGKM0235E Cannot obtain the progress of the running process.

Explanation: Cannot obtain the progress of the running backup or restore process.

System action: Failed to get the progress of the running process.

Administrator response: Check the log entries to find out the reason for the failure. Make appropriate changes. Then, try the operation again.

CTGKM0236E Cannot obtain the progress state of the running process.

Explanation: Cannot obtain the progress state of the running backup or restore process.

System action: Failed to get the progress state of the running process.

Administrator response: Check the log entries to find out the reason for the failure. Make appropriate changes. Then, try the operation again.

CTGKM0237E Cannot obtain the process result of the completed process.

Explanation: Cannot obtain the process result of the completed backup or restore process.

System action: Failed to get the process result of the completed process.

Administrator response: Check the log entries to find out the reason for the failure. Make appropriate changes. Then, try the operation again.

CTGKM0238E Cannot create backup.

Explanation: Cannot initiate the backup process.

System action: Failed to initiate the backup process.

Administrator response: Check the log entries to find out the reason for the failure. Make appropriate changes. Then, try the operation again.

CTGKM0239E Cannot restore from backup.

Explanation: Cannot initiate the restore from backup process.

System action: Failed to initiate the restore from backup process.

Administrator response: Check the log entries to find out the reason for the failure. Make appropriate changes. Then, try the operation again.

CTGKM0240E Cannot delete backup.

Explanation: Cannot delete the backup process.

System action: Failed to delete the backup process.

Administrator response: Check the log entries to find out the reason for the failure. Make appropriate changes. Then, try the operation again.

CTGKM0242E Failed to create backup to *backup_file* **.**

Explanation: Cannot back up the IBM Security Key Lifecycle Manager system.

System action: Backup operation failed to back up the IBM Security Key Lifecycle Manager system.

Administrator response: Check the log entries to find out the reason for the failure. Make appropriate changes. Then, try the operation again.

CTGKM0244E Failed to restore from *backup_file* **.**

Explanation: Cannot restore the IBM Security Key Lifecycle Manager system.

System action: Restore operation failed to restore the IBM Security Key Lifecycle Manager system.

Administrator response: Check the log entries to find out the reason for the failure. Make appropriate changes. Then, try the operation again.

CTGKM0245E The key name specified is not known.

Explanation: You might have incorrectly specified the key name value. Alternatively, IBM Security Key Lifecycle Manager database might not be available or an internal component such as the key server component returned an error or was not available.

System action: The operation fails.

Administrator response: Ensure that your value for a key name is valid and is in the keystore. Ensure that the IBM Security Key Lifecycle Manager database is available. If an internal component failed, you might restart the IBM Security Key Lifecycle Manager server. Then, try the operation again.

CTGKM0246E Cannot identify the next key to be used from this key group.

Explanation: An attempt failed to determine the next key to be used from this key group. There might be more information in the message that describes the problem.

System action: The operation fails.

Administrator response: Additional information might guide your response. Make appropriate changes. Then try the operation again.

CTGKM0247E Cannot retrieve future write defaults.

Explanation: An error occurred while retrieving a list of future write defaults. There might be more information in the message that describes the problem.

System action: The operation fails.

Administrator response: Additional information might guide your response. Make appropriate changes. Then, try the operation again.

CTGKM0248E Cannot retrieve current system default.

Explanation: An internal component such as the key server component returned an error or was not available. The SKLMConfig.properties file might be write protected. This message might provide additional information.

System action: System default retrieval fails.

Administrator response: Ensure that the SKLMConfig.properties file is write enabled. You might restart the IBM Security Key Lifecycle Manager server. Then, try the operation again. Additional information in this message might guide your response.

CTGKM0249E Cannot add future write default.

Explanation: You might have specified a value that is not valid for a future write default. Alternatively, IBM Security Key Lifecycle Manager database might not be available or an internal component such as the key server component returned an error or was not available.

System action: The operation fails.

Administrator response: Ensure that your values for a future write default are valid. Ensure that the IBM Security Key Lifecycle Manager database is available. If an internal component failed, you might restart the IBM Security Key Lifecycle Manager server. Then, try the operation again.

CTGKM0250E Cannot delete future write default.

Explanation: IBM Security Key Lifecycle Manager could not delete a future write default. There might be additional information in this message.

System action: The future write default is retained in the IBM Security Key Lifecycle Manager database.

Administrator response: Additional information in this message might guide your response. Make appropriate changes. Then, try the operation again.

CTGKM0251E Future write default's effective date cannot be later than the certificate expiration date.

Explanation: If the certificate expires before the future write default's effective date, then this future write default will not be effective.

System action: Adjust the future write default's effective date.

Administrator response: Extend the certificate expiration date if necessary.

CTGKM0252E Algorithm type for *backup_file* **application is not valid.**

Explanation: This is not a valid algorithm type for the intended application. For IKEV2SERVER and IKVE2CLIENT, the only algorithm type supported is ECDSA. Cannot import this key or certificate for this application.

System action: Import operation fails.

Administrator response: Generate an ECDSA type of certificate or key for IKVE2SERVER or IKEV2CLIENT and then try the import operation.

CTGKM0253E Device serial number for LTO base device *VALUE_0* **must be either 10, 12 or 24 characters long, contain valid characters, and cannot have leading or trailing whitespace.**

Explanation: The device serial number for a device must be either 10, 12 or 24 characters in length and contain only allowed characters.

System action: The operation fails.

Administrator response: Ensure that the device serial number you enter is 10, 12 or 24 valid characters in length and contains alphanumeric, period, dash, semicolon, and underscore characters, or a space that is not in the first or last position. Additional information in this message might also guide your response. Correct the problem and try the operation again.

CTGKM0255E Cannot delete key. Key is the last active member of a device.

Explanation: The current active member key of a device or group cannot be deleted.

System action: The key is not deleted.

Administrator response: Additional information might guide your response. Make appropriate changes. Then, try the operation again.

CTGKM0256E Incorrect key size *VALUE_0*.

Explanation: This is not a valid key size type for the intended application. For IKEV2SERVER and IKVE2CLIENT, the only key size supported is 521 bits. Cannot import this key or certificate for this application.

System action: Import operation fails.

Administrator response: Generate correct size of ECDSA type of certificate or key for IKVE2SERVER or IKEV2CLIENT and then try the import operation.

CTGKM0257E Unable to accept the Pending Client Device Communication Certificate.

Explanation: The certificate name might already exist, the certificate material might already exist under a different name, or the IBM Security Key Lifecycle Manager database might not be available. There might be additional information.

System action: Accept operation fails.

Administrator response: Determine if you specified a unique certificate name. Alternatively, you might need to confirm that the database is available. Additional information in this message might also guide your response. Make corrections. Then, try the operation again.

CTGKM0258E Unable to reject the Pending Client Device Communication Certificate.

Explanation: The certificate might not exist, or the IBM Security Key Lifecycle Manager database might not be available. There might be additional information.

System action: Reject operation fails.

Administrator response: Refresh the page to determine if the certificate still exists in the pending list. Alternatively, you might need to confirm that the database is available. Additional information in this message might also guide your response. Make corrections. Then, try the operation again.

CTGKM0259E Unable to gather additional information on the selected Pending Client Device Communication Certificate.

Explanation: The certificate information was not available from the IBM Security Key Lifecycle Manager database.

System action: View operation fails.

Administrator response: Ensure that the IBM Security Key Lifecycle Manager database is available. Correct any database server runtime errors that you identify. Then, try the operation again.

CTGKM0260E Keystore internal error occurred.

Explanation: An internal error occurred while creating or loading the keystore.

System action: Keystore operation fails.

Administrator response: Verify that IBM Security Key Lifecycle Manager is initialized correctly.

CTGKM0261E Alias {0} already exists.

Explanation:

System action: Keystore operation fails.

Administrator response: Specify a different alias.

CTGKM0262E The certificate expiration period cannot be later than *VALUE_0* years. This value can be changed by use of the configuration parameter called maximum.keycert.expiration.period.in.years.

Explanation: The certificate expiration period cannot be later than the configured years. This value can be changed by use of the configuration parameter called maximum.keycert.expiration.period.in.years.

System action: Adjust the maximum.keycert.expiration.period.in.years if necessary.

Administrator response: Adjust the maximum.keycert.expiration.period.in.years if necessary.

CTGKM0263E You have successfully restored your system from *backup_file* You will be required to manually restart the server as automatic restart failed. Please refer to the Backup and restore section of IBM Security Key Lifecycle Manager documentation, on how to start and stop the server on distributed systems.

Explanation: Backup was successfully restored but restart failed.

CTGKM0400E You must specify a value for: *VALUE_0*

Explanation: You attempted to update a key value, but you specified a null parameter.

System action: The key is not updated.

Administrator response: Enter a valid parameter value and try the update operation again.

CTGKM0401E The alias name prefix (3 characters) is not alphabetic: *VALUE_0*

Explanation: An alias prefix for a key must be three characters in length, and the characters must be alphabetic, which are the characters A-Z case insensitive.

System action: The key or keys are not created.

Administrator response: Specify a three character value for the key alias using the alphabetic characters A-Z, which are case insensitive.

CTGKM0402E The alias range (*VALUE_0* **,** *VALUE_1* **) is not valid.**

Explanation: The first hexadecimal number in an alias range must smaller than the last hexadecimal number that you specify.

System action: The keys are not created in the specified range.

Administrator response: Ensure that the values that you specified for the alias range are valid hexadecimal numbers, and that the initial number is less than the final number in the range. For example, specify a range such as xyz01-fff. Then, try the operation again.

CTGKM0403E The alias does not contain all alphabetic characters: *VALUE_0*

Explanation: The alias value must contain only alphabetic characters.

System action: The key or keys are not created.

Administrator response: Ensure that the value that you specified for the key alias contains only alphabetic characters. Then, try the operation again.

CTGKM0404E The alias range length is too large.

Explanation: The alias range that you specified exceeds a IBM Security Key Lifecycle Manager limit.

System action: The key or keys are not created.

Administrator response: Specify a smaller alias range. Then, try the operation again.

CTGKM0405E The alias range end value must be greater than or equal to the alias range start value.

Explanation: The last hexadecimal number in an alias range must greater than or equal to the first hexadecimal number that you specify.

System action: The keys are not created in the specified range.

Administrator response: Ensure that the values that you specified for the alias range are valid hexadecimal numbers, and that the final number is greater than or equal to the initial number in the range. For example, specify a range such as xyz01-fff. Then, try the operation again.

CTGKM0406E The alias range is too large.

Explanation: The alias range that you specified exceeds a IBM Security Key Lifecycle Manager limit.

System action: The keys are not created in the specified range.

Administrator response: Specify a smaller alias range. Then, try the operation again.

CTGKM0407E The alias range string contains a non-hexadecimal value.

Explanation: The numbers that specify an alias range must be hexadecimal numbers.

System action: The keys are not created in the specified range.

Administrator response: Specify an alias range using only hexadecimal values. Then, try the operation again.

CTGKM0408E Cannot locate certificate chain with alias *VALUE_0*

Explanation: The alias that you specified for a certificate does not contain a certificate chain.

System action: The operation fails.

Administrator response: Ensure that the alias that you specified contains a certificate chain. Then, try the operation again.

CTGKM0409E Check the file name. Cannot export the key to *VALUE_0***.**

Explanation: There is a write error to the file on which an export operation was attempted. There is possibly an error in the relative or full path, or the name of the file that IBM Security Key Lifecycle Manager creates to store private keys.

System action: The key is not exported.

Administrator response: Ensure that you correctly specified the relative or full path, and the name of the file to store the exported keys. If you do not specify a path name, the value of the SKLM_HOME directory is used. Then, try the operation again.

CTGKM0410E Error occurred while exporting the key to output stream.

Explanation: The certificate export operation did not write to the file on which an export operation was attempted. There is possibly an error in the relative or full path, or the name of the file that IBM Security Key

Lifecycle Manager creates to store the certificate.

System action: The certificate is not exported.

Administrator response: Ensure that the path and file name are correct. Then, try the operation again.

CTGKM0411E ***VALUE_0*** **is not a secret key entry in the keystore.**

Explanation: The specified secret key is not in the keystore.

System action: An operation fails, such as exporting a secret key.

Administrator response: You might use the tklmKeyList command to view the keys contained in the keystore. Correct any errors in your specification. Then, try the operation again.

CTGKM0412E ***VALUE_0*** **is not a private key entry in the keystore.**

Explanation: The specified private key is not in the keystore.

System action: The operation fails, such as exporting a private key to a PKCS12 file.

Administrator response: You might use the tklmKeyList command to view the private keys contained in the keystore. Correct any errors in your specification. Then, try the operation again.

CTGKM0413E **Unsupported private key algorithm:** *VALUE_0*

Explanation: This is not one of the key algorithms that IBM Security Key Lifecycle Manager supports.

System action: The key operation fails. For example, you might be trying to import a private key that does not match a supported RSA or DSA algorithm.

Administrator response: Use a different key that complies with an asymmetric key algorithm that IBM Security Key Lifecycle Manager supports.

CTGKM0414E **File size is zero.**

Explanation: The specified key file is empty, from which you are attempting to import a key. There is no data in the file.

System action: The operation fails.

Administrator response: Ensure that the key file is correctly populated, and that you have a valid key file from a trusted source. Then, try the operation again.

CTGKM0415E **Cannot find the file** *VALUE_0*

Explanation: The key file was not found during a key import operation.

System action: The key import operation fails.

Administrator response: Ensure that you specified the correct path and filename. Then, try the operation again.

CTGKM0416E **Cannot find the** ***VALUE_0*** **in the specified group.**

Explanation: The group member that you specified is not in the target group.

System action: The operation fails.

Administrator response: Ensure that your specifications of both the group member and the group are correct. You might first need to add the member to the group. Then, try the operation again.

CTGKM0417E **Cannot delete a key from a device group.**

Explanation: The IBM Security Key Lifecycle Manager database stores group entries for keys in a key group type of group.

System action: The group entry delete operation fails.

Administrator response: Change your specification of the group type to a value of key group. Then, try the operation again.

CTGKM0419E **Entry** ***VALUE_0*** **does not belong to any group.**

Explanation: The IBM Security Key Lifecycle Manager database stores entries in a group with a type of key group. The entry that you are attempting to delete was not found in any type of group.

System action: The operation fails.

Administrator response: Ensure that your specification of the entry uuid and the group name are correct. Then, try the operation again.

CTGKM0420E **Key group is empty.**

Explanation: There are no keys in the group. This is an internal message that the key server might issue to a log.

System action: The key operation fails.

Administrator response: You might need to add keys to the key group. The change is effective immediately. However, you might need to restart the IBM Security Key Lifecycle Manager server. Then, try the operation again.

CTGKM0421E Error occurred while encrypting the key.

Explanation: Encryption failed during a secret key export operation to a file. There might be a problem with the encryption provider.

System action: The export operation fails.

Administrator response: Collect any information that might be in the audit log. You might need to contact IBM Support.

CTGKM0422E Error occurred while encoding data in PKCS12 format.

Explanation: An exception occurred in internal processes.

System action: The private key and certificate are not encoded.

Administrator response: Collect any information that might be in the audit log and contact IBM Support.

CTGKM0423E Error occurred while verifying the key and certificate.

Explanation: An exception occurred in internal processes.

System action: The certificate request that you submitted to a CA and the certificate that returned, do not match. This might be an internal processing error.

Administrator response: Collect any information that might be in the audit log and contact IBM Support.

CTGKM0424E Error occurred while decrypting the secret key.

Explanation: An exception occurred in internal processes.

System action: The secret key was not decrypted. This might be an internal processing error.

Administrator response: Collect any information that might be in the audit log and contact IBM Support.

CTGKM0425E The alias prefix does not have 3 characters.

Explanation: An alias prefix for a key must be 3 characters in length, and the characters must be alphabetic.

System action: The key or keys are not created.

Administrator response: Ensure that the value that you specified for the key alias contains 3 alphabetic characters. Then, try the operation again.

CTGKM0426E Cannot delete the certificate as it is associated with a private key entry.

Explanation: You used the tklmCertDelete command to delete a certificate that has a private key associated with the certificate.

System action: The certificate was not deleted.

Administrator response: Use the tklmKeyDelete command to delete the key. The alias for the certificate and the key are the same, and are stored internally as a single entry.

CTGKM0427E Group name is longer than 64 characters: *VALUE_0*

Explanation: The limit for a group name is 64 characters.

System action: The operation fails.

Administrator response: Specify a shorter group name that meets the character limit. Then, try the operation again.

CTGKM0428E Default key group cannot be deleted.

Explanation: One or more devices use this key group as the default from which to obtain keys.

System action: The key group delete operation fails.

Administrator response: Ensure that no device uses this key group as the default from which to obtain keys. You might select another key group as the default. Then, try the operation again.

CTGKM0429E Alias range does not start with 3-letter prefix.

Explanation: The alias range prefix must be 3 characters in length.

System action: The operation fails.

Administrator response: Specify a prefix that contains 3 characters for the alias range. Then, try the operation again.

CTGKM0430E Alias range does not have a hexdecimal range separated by dash.

Explanation: A dash is the required separator between hexadecimal numbers in an alias range.

System action: The range of keys is not created.

Administrator response: Specify the alias range using a dash as the separator. Then, try the operation again.

CTGKM0431E Starting with version 2, this message is deprecated. aliasOne attribute must be specified for DS8000 device.

Explanation: Starting with version 2, this message is deprecated. aliasOne is the required attribute for DS8000 device.

System action: The device is not created.

Administrator response: Specify the aliasOne attribute. Then, try the operation again.

CTGKM0432E Rollover task for type *VALUE_0* **is already scheduled on effective date:** *VALUE_1*

Explanation: Effective date is unique for each rollover type

System action: The rollover is not created.

Administrator response: Specify different rollover date and type. Then, try the operation again.

CTGKM0433E Two-letter country code: *VALUE_0* **is not valid.**

Explanation: The specified value does not have 2 letters.

System action: The certificate or certificate request command fails.

Administrator response: Specify a different country code. Then, try the command again.

CTGKM0434E Cannot delete the key because it is the default symmetric key: *VALUE_0*

Explanation: The specified key is a default symmetric key.

System action: The delete operation failed.

Administrator response: Specify a different key and try the delete operation again.

CTGKM0435E All keys in the key group are used. There is no key available to use.

Explanation: All keys in the key group are used. There is no key available to use.

System action: The operation to get the next key failed

Administrator response: Add a new key to the key group and try to get a key again.

CTGKM0436E The key group cannot be deleted, but the key group members are deleted successfully.

Explanation: The key group is not deleted.

System action: The key group delete operation failed.

Administrator response: Examine the logs for information about the error. Make necessary corrections. Then, try the operation again. If the problem still exists, you might need to contact IBM Support.

CTGKM0437E The key group cannot be deleted, but some members may have been deleted.

Explanation: The key group is not deleted.

System action: The key group delete operation failed.

Administrator response: Examine the logs for information about the error. Make necessary corrections. Then, try the operation again. If the problem still exists, you might need to contact IBM Support.

CTGKM0438E The last key in the default key group cannot be deleted.

Explanation: The key is not deleted.

System action: The key delete operation fails.

Administrator response: The default key group needs to include at least one key.

CTGKM0439E Cannot delete the last key in the key group that is associated with a device.

Explanation: The key is not deleted.

System action: The key delete operation fails.

Administrator response: The key group associated with a device needs to include at least one key.

CTGKM0440E Cannot delete the last key in the key group *VALUE_0* **that is associated with a scheduled rollover.**

Explanation: The key is not deleted.

System action: The key delete operation fails.

Administrator response: The key group associated with a rollover need to have at least one key.

CTGKM0441E Key Export only supports IBMJCE and IBMJCEFIPS providers.

Explanation:

System action:

Administrator response:

CTGKM0500E A keystore already exists in IBM Security Key Lifecycle Manager.

Explanation: Only one IBM Security Key Lifecycle Manager keystore can exist.

System action: The keystore add operation fails.

Administrator response: Use the existing IBM Security Key Lifecycle Manager keystore. You cannot add an additional keystore. If you must use a new keystore, you must first remove the existing keystore and add a new one. In a running production environment, do not modify the keystore name. If you must modify the keystore name prior to production, ensure that you have a complete, current backup of your IBM Security Key Lifecycle Manager configuration.

CTGKM0501E No password.

Explanation: You must specify a password for a IBM Security Key Lifecycle Manager keystore. The password must at least 6 characters in length.

System action: The keystore operation fails.

Administrator response: Specify a password for the IBM Security Key Lifecycle Manager keystore. For example, type a value such as my6pwd.

CTGKM0502E Keystore name must be specified.

Explanation: You must specify a name for a IBM Security Key Lifecycle Manager keystore. IBM Security Key Lifecycle Manager uses this name in the IBM Security Key Lifecycle Manager database as a descriptive alias to identify the keystore.

System action: The keystore operation fails.

Administrator response: Specify a name as a descriptive alias for the IBM Security Key Lifecycle Manager keystore. For example, type mySKLMKeystore.

CTGKM0503E Duplicate keystore name.

Explanation: A duplicate name exists in the IBM Security Key Lifecycle Manager database for the value that you specified as a descriptive alias for a IBM Security Key Lifecycle Manager keystore.

System action: The keystore operation fails.

Administrator response: Type a unique value for the name of the IBM Security Key Lifecycle Manager keystore. For example, type mySKLMKeystore.

CTGKM0504E Error occurred while creating the keystore:

Explanation: You might not have correctly specified the values needed to add a file-based keystore.

System action: The keystore operation fails.

Administrator response: Ensure that you specified a unique value for the descriptive alias, the path and file name of the keystore, or the keystore type. For a RACF keystore, you might have to specify the user ID or password differently. After modifying your entries, try the operation again.

CTGKM0505E Error occurred while loading the keystore:

Explanation: This error occurs when you attempt to read data from an existing keystore.

System action: The keystore operation fails.

Administrator response: The data in the keystore might be corrupt, or the path specification or password value might be incorrect. If the keystore is corrupt, use a backup copy. Otherwise, respecify the path or password values, and try the operation again.

CTGKM0506E Internal database operation error.

Explanation: An unexpected database error occurred.

System action: The database operation did not complete as expected.

Administrator response: Collect any information that might be in the audit log and contact IBM Support.

CTGKM0507E Validation error on input:

Explanation: The value that you provided did not match an expected value.

System action: The value was not written or retained.

Administrator response: Ensure that the value you provided is valid. Then, try the operation again.

CTGKM0508E Failed to execute the command:

Explanation: There is an error in the values provided for the tklmKeyStoreList command.

System action: The command operation fails.

Administrator response: Ensure that the command syntax and values are correct. Then, try the command again.

CTGKM0509E Password cannot be shorter than 6 characters.

Explanation: The password strength rule requires a length of six or more characters.

System action: The keystore operation fails.

Administrator response: Specify a password that is at least six characters in length, and then try the operation again.

CTGKM0510E Keystore file path name is not specified.

Explanation: You must specify a complete path to the keystore file.

System action: The keystore operation fails.

Administrator response: Specify a complete path to the keystore file, either as an absolute or relative path. Then, try the operation again. As a relative path, for example, you might specify a file name in an existing directory. IBM Security Key Lifecycle Manager appends the value of SKLM_HOME as the relative path.

CTGKM0512E Keystore type is not valid *KEYSTORE_TYPE*

Explanation: You must specify a keystore type that IBM Security Key Lifecycle Manager supports.

System action: The keystore operation fails.

Administrator response: Specify a supported keystore type. Then, try the operation again.

CTGKM0513E Cannot find MBean:

Explanation: This error might occur if you install IBM Security Key Lifecycle Manager and then move or delete some of its files.

System action: Internal descriptive files are not found.

Administrator response: Do not move or delete files from their installed locations without explicit, authorized instructions. You might need to contact IBM Support.

CTGKM0514E Target application usage must be specified.

Explanation: Usage specifies how the certificate is used with a target application, either for secure communication using an SSL server protocol, or for communication with a device.

System action: The certificate operation fails.

Administrator response: Specify a valid value for the certificate usage. Then, try the operation again.

CTGKM0515E Certificate name must be specified.

Explanation: The name uniquely identifies the certificate within the keystore.

System action: The certificate operation fails.

Administrator response: Specify the name for the certificate. Then, try the operation again.

CTGKM0516E Common name must be specified.

Explanation: The common name (cn) is part of the unique identification for the certificate. For example, the value of cn is used in the subject name for a certificate, which can identify whether a certificate that is being imported matches an original certificate request.

System action: The operation fails.

Administrator response: Specify the common name for the certificate. Then, try the operation again.

CTGKM0517E Cannot form a valid X.500 name.

Explanation: One or more of the values for the subject fields is not valid to generate a valid X.500 directory name, which must be unique to identify a self-signed certificate as an entry for a global directory service.

System action: The X.500 name is not created.

Administrator response: Ensure that you provided correct values for the subject fields that comprise an X.500 name, such that a certificate can be unambiguously identified. For example, ensure that the value for the cn field is unique, and that values are also complete in other fields.

CTGKM0518E Keystore does not exist.

Explanation: A keystore must exist to contain the certificate. You might have entered an incorrect keystore name, or used a command that requires a value for the keystore, before creating the keystore.

System action: The certificate operation fails.

Administrator response: Specify a valid, existing keystore. You might need to create a keystore using either the tklmKeystoreAdd command, or the graphical user interface. Then, try the operation again.

CTGKM0520E Not the supported usage: *VALUE_0*

Explanation: You specified an unsupported value for the target usage. The target application specifies how the certificate is used, either for secure communication using a SSL protocol, or for communication with a device such as a tape drive.

System action: The certificate operation fails.

Administrator response: Specify a valid target application for which the certificate is used. Then, try the operation again.

CTGKM0521E Unsupported certificate format: *VALUE_0*

Explanation: You specified an value that is not valid for the certificate format. IBM Security Key Lifecycle Manager supports either a base64 and DER format for certificates.

System action: The certificate operation fails.

Administrator response: Specify either base64 or DER as the certificate format. Then, try the operation again.

CTGKM0522E No certificate information.

Explanation: During a certificate operation, the certificate information was missing or null.

System action: The certificate import operation fails.

Administrator response: Ensure that you identified the certificate with a valid name. For example, run the tklmCertList command to find the certificate name. Then, try the operation again.

CTGKM0523E No keystore information.

Explanation: During a certificate import or keystore delete operation, the keystore parameter was null. The value of the keystore is missing.

System action: The certificate operation fails.

Administrator response: Determine whether you identified the keystore with a valid name. For example, run the tklmKeystoreList command to find the keystore name. Then, try the operation again.

CTGKM0524E Keystore name or uuid must be specified.

Explanation: During a certificate import or keystore delete operation, the keystore parameter was null. The value of the keystore is missing.

System action: The certificate operation fails.

Administrator response: Determine whether you identified the keystore with a valid name. For example, run the tklmKeystoreList command to find the keystore name. Then, try the operation again.

CTGKM0525E Parameter value(s) are not valid.

Explanation: During general validation, at least one value was not found for a required parameter for the tklmCertCreate or tklmCertUpdate command.

System action: The certificate operation fails.

Administrator response: Specify the missing value or values. Then, try the operation again.

CTGKM0526E Certificate file name must be specified.

Explanation: A certificate file name must be specified for the tklmCertExport or the tklmCertImport command.

System action: The certificate operation fails.

Administrator response: Specify the file name of the certificate. Then, try the operation again.

CTGKM0527E Certificate validity value *VALUE_0* **is not valid**

Explanation: The length of time in days that you specified for the certificate does not fall within the expected range. The value must be 1 or greater.

System action: The certificate operation fails.

Administrator response: Specify a valid length of time in days during which the certificate can be used. Then, try the operation again.

CTGKM0528E Keystore name cannot exceed 64 characters.

Explanation: For the tklmKeystoreAdd command, you specified a keystore name that exceeds a limit of 64 characters.

System action: The certificate operation fails.

Administrator response: Specify a keystore name that is 64 characters or less in length. Then, try the operation again.

CTGKM0529E An error occurred generating certificate request.

Explanation: An exception occurred in internal processes.

System action: The certificate request is not created.

Administrator response: Verify the parameter values for the certificate request. Then, try the operation again.

CTGKM0530E Cannot find the certificate.

Explanation: The value of the alias or uuid was not found when you attempted an operation such as deleting, exporting, or updating a certificate.

System action: The certificate operation fails.

Administrator response: Specify a valid alias or uuid for the target certificate. Then, try the operation again.

CTGKM0531E uuid must be specified.

Explanation: A value is required for the uuid parameter for a delete operation.

System action: The group delete operation fails.

Administrator response: Ensure that you specified the correct value for the uuid of the group. You might use the tklmGroupList command to verify values. Then, try the operation again.

CTGKM0532E Compromise date cannot be later than today.

Explanation: The date on which you specify that a key or a certificate is compromised cannot be a future date.

System action: The operation fails.

Administrator response: Specify a date that is not in the future, and then try the operation again.

CTGKM0533E Activation date, retirement date, expiration date and destroy date are not synchronized.

Explanation: There is a mismatch in dates between several parameters, which must follow each other in time. A destroy date value must occur after an expiration date value, for example.

System action: The update certificate operation fails.

Administrator response: Enter values for dates that do not conflict in their sequence on the calendar. Then, try the operation again.

CTGKM0534E Cannot reset activation date, old activation date has passed:

Explanation: If the date of activation is older than the current date, you cannot reset a new value for activation.

System action: The operation fails.

Administrator response: Use a different certificate. The activation date of this certificate cannot be reset.

CTGKM0535E Cannot reset retirement date, old retirement date has passed:

Explanation: If the date of retirement is older than the current date, you cannot reset a new value for retirement.

System action: The operation fails.

Administrator response: Use a different certificate. This certificate is retired, and the retirement date cannot be reset.

CTGKM0536E Cannot reset destroy date, old destroy date has passed:

Explanation: If the destroy date is older than the current date, you cannot reset a new value for the destroy date.

System action: The operation fails.

Administrator response: Use a different certificate. The destroy date of this certificate cannot be reset.

CTGKM0537E Cannot set state to be *VALUE_0* **, current state is** *VALUE_1* **.**

Explanation: The current state of the certificate cannot be reset to the state that you specified. For example, you cannot set the state of an active certificate to a pre-active state.

System action: The operation fails.

Administrator response: Specify a different state for the certificate. Then, try the operation again.

CTGKM0538E Error setting new state: *VALUE_0* **, certificate is already compromised.**

Explanation: A certificate that is in a compromised state cannot be set to an earlier state.

System action: The operation fails.

Administrator response: Specify a subsequent state. Then, try the operation again. Alternatively, you might need to use an uncompromised certificate.

CTGKM0539E Error setting new state: *VALUE_0* **, certificate is not compromised.**

Explanation: A certificate must be in a compromised state before a subsequent state can be set.

System action: The operation fails.

Administrator response: First, change the state of the certificate to compromised. Then, try the operation again.

CTGKM0540E Certificate with alias *VALUE_0* **already exists in keystore.**

Explanation: There is already a certificate in the keystore with the same alias as the one you are attempting to import.

System action: The operation fails.

Administrator response: Specify a different alias. Then, try the operation again.

CTGKM0541E uuid and certificate attributes must be specified.

Explanation: There might be an error in the uuid value for a certificate.

System action: The operation fails.

Administrator response: Ensure that the uuid value of the certificate is valid. Then, try the operation again.

CTGKM0542E Not a *VALUE_0* **encoded certificate file. Make sure to use the right file and it is not tampered or corrupted.**

Explanation: The import operation determined that the certificate file is not correctly encoded. It must be DER or base64 format.

System action: The operation fails.

Administrator response: Specify a certificate that has a DER or base64 format. Then, try the operation again.

CTGKM0543E An error occurred importing certificate: {0}

Explanation: The certificate file might not exist, or you might have made an error in specifying the file name of the certificate.

System action: The operation fails.

Administrator response: Determine whether the certificate path and file name are correct, and whether the file is corrupt. Correct the problems. Then, try the operation again.

CTGKM0544E Cannot retrieve the certificate from keystore.

Explanation: Your entries might have an error in specifying the certificate alias. Alternatively, the keystore might not contain the target certificate.

System action: The operation fails.

Administrator response: Determine whether you specified the certificate alias correctly. Alternatively, you might use the tklmKeystoreList command to determine which certificates the keystore contains. Then, try the operation again.

CTGKM0545E An error occurred exporting certificate.

Explanation: The file that you specified might be read-only, or you do not have permission to write the file to a specific location.

System action: The export operation fails.

Administrator response: Ensure that the file is write enabled, and that your permissions are valid. Then, try the export operation again.

CTGKM0546E Expiration date cannot be early than activation date: *VALUE_0*

Explanation: The expiration date of a certificate must occur later in time than the activation date.

System action: The operation fails.

Administrator response: Specify an expiration date that is later than the activation date. Then, try the operation again.

CTGKM0547E Expiration date cannot be later than retirement date: *VALUE_0*

Explanation: The expiration date of a certificate must occur earlier in time than the retirement date.

System action: The operation fails.

Administrator response: Specify an expiration date that is earlier than the retirement date. Then, try the operation again.

CTGKM0548E Expiration date cannot be later than destroy date: *VALUE_0*

Explanation: The expiration date of a certificate must occur earlier in time than the destroy date.

System action: The operation fails.

Administrator response: Specify an expiration date that is earlier than the destroy date. Then, try the operation again.

CTGKM0549E Subject name of the certificate does not match subject name in certificate request.

Explanation: The subject name of the certificate that returned from a Certificate Authority does not match the subject name in the original certificate request.

System action: The import operation fails.

Administrator response: Correct the file name or alias specification. Then, try the operation again.

CTGKM0550E Input value cannot be an empty string for parameter *VALUE_0*

Explanation: The entry for the attribute name must not be blank or a space.

System action: The operation fails.

Administrator response: Specify one or more valid characters for this entry. Then, try the operation again.

CTGKM0551E Cannot find keystore provider for keystore type *VALUE_0*

Explanation: The keystore type does not match the set of supported providers.

System action: The operation fails.

Administrator response: Specify a supported keystore type. Then, try the operation again.

CTGKM0552E *VALUE_0* **type keystore is not supported.**

Explanation: The keystore type is not in the set of supported providers.

System action: The operation fails.

Administrator response: Specify a supported keystore type. Then, try the operation again.

CTGKM0555E Default property cannot be deleted: *VALUE_0*

Explanation: You attempted to delete a property from the SKLMConfig.properties file that IBM Security Key Lifecycle Manager uses as a default property.

System action: The configuration operation fails.

Administrator response: Ensure that the property you intend to delete is a customized property, or a property other than a default property. Then, try the delete operation again.

CTGKM0556E Cannot find the property in configuration file: *VALUE_0*

Explanation: Using the tkmlConfigDeleteEntry command, you attempted to delete a property that does not exist in the properties file.

System action: The configuration operation fails.

Administrator response: Ensure that the property exists. You might use the tklmConfigList command to list the contents of the IBM Security Key Lifecycle Manager configuration file. Specify the target property and try the delete operation again.

CTGKM0557E Cannot delete config.keystore.name property in configuration file

Explanation: You attempted to delete the config.keystore.name property from the SKLMConfig.properties file. IBM Security Key Lifecycle Manager must use this property to identify the keystore.

System action: The configuration operation fails.

Administrator response: You might change the value of the config.keystore.name property, but you cannot delete the property itself. In a running production environment, do not modify the keystore name. If you must modify the keystore name prior to production, ensure that you have a complete, current backup of your IBM Security Key Lifecycle Manager configuration.

CTGKM0558E Keystore named *KEYSTORE_NAME* **cannot be found. Another user may have renamed the keystore. Try the operation again.**

Explanation: While specifying a tklmKeystoreUpdate or tklmKeystoreDelete command, you did not specify the correct value for the storeName parameter to identify the existing IBM Security Key Lifecycle Manager keystore.

System action: The operation fails.

Administrator response: Specify the correct value for the existing keystore. In the SKLMConfig.properties file, the config.keystore.name property specifies the value of the IBM Security Key Lifecycle Manager keystore. You might also run the tklmKeyStoreList command to identify the keystore name.

CTGKM0559E Group name and type must be specified.

Explanation: The group name is missing, or the group type is not valid.

System action: The operation fails.

Administrator response: Specify a value for the group name and a valid group type. Then, try the operation again.

CTGKM0560E *VALUE_0* **cannot be null.**

Explanation: A value other than a space or blank is required.

System action: The operation fails.

Administrator response: Specify a supported value that is not a blank or a space. Then, try the operation again.

CTGKM0561E Unsupported group type: *VALUE_0*

Explanation: The value that you specified for the group type is not supported.

System action: The operation fails.

Administrator response: Specify a supported value for a group type, such as keygroup. Then, try the operation again.

CTGKM0562E Cannot find the group: *VALUE_0*

Explanation: The value of the group name that you specified does not match an existing group. The group name might be incorrect, or the group might not exist in the type of group that you specified.

System action: The operation fails.

Administrator response: Specify a group that exists in the group type that you intend to use. You might use the tklmGroupList command to verify that the group exists in an intended type. Then, try the operation again.

CTGKM0563E Cannot add a key to a device group.

Explanation: The value of the group type is incorrect.

System action: The operation fails.

Administrator response: Specify keygroup as the group type. Then, try the operation again.

CTGKM0564E Cannot identify the entry: *VALUE_0*

Explanation: The value of the entry that you specified does not match an existing certificate, key, or device. The certificate, key, or device identifier might be incorrect, or the identifier might not exist in the entry type that you specified.

System action: The operation fails.

Administrator response: Specify an entry that exists in the entry type that you intend to use. If you are deleting an entry from a group, you might first use the tklmKeyList command to verify that the entry exists in the intended type. Then, try the operation again.

CTGKM0565E Cannot find the key: *VALUE_0*

Explanation: The key value that you specified does not match an existing key.

System action: The operation fails.

Administrator response: Specify a key that exists in the type that you intend to use. If you are deleting a key from a group, you might first use the tklmGroupList or the tklmKeyList command to verify that the key exists. Then, try the operation again.

CTGKM0566E Cannot add a certificate to a device group.

Explanation: The value of the group type is incorrect.

System action: The operation fails.

Administrator response: Specify keygroup as the group type. Then, try the operation again.

CTGKM0567E Cannot find the certificate: *VALUE_0*

Explanation: The certificate value that you specified does not match an existing certificate.

System action: The operation fails.

Administrator response: Specify a certificate that exists. You might first use the tklmCertList command to verify that the certificate exists. Then, try the operation again.

CTGKM0569E Cannot find the device: *VALUE_0*

Explanation: The device value that you specified does not match an existing device.

System action: The operation fails.

Administrator response: Specify a device that exists. You might first use the tklmDeviceList command to verify that the device exists. Then, try the operation again.

CTGKM0570E Failed to add group entry.

Explanation: The entry was not added to the group. This might be an internal error.

System action: The operation fails.

Administrator response: The audit log might contain information about the error. Collect the information and contact IBM Support.

CTGKM0571E File already exists: *VALUE_0*

Explanation: The file that you are attempting to export matches the name of an existing file.

System action: The operation fails.

Administrator response: Specify a different path and file name. Then, try the operation again.

CTGKM0580E Keystore name and certificate alias must be specified

Explanation: Your command did not specify the keystore name or the certificate alias.

System action: The operation fails.

Administrator response: Specify the keystore name and the certificate alias correctly. Alternatively, you might use the tklmKeystoreList command to determine which certificates the keystore contains. Then, try the operation again.

CTGKM0581E Internal key server exception.

Explanation: The IBM Security Key Lifecycle Manager data store returned an error when attempting to delete a certificate.

System action: The operation fails.

Administrator response: The audit log might contain information about the error. Collect the information and contact IBM Support.

CTGKM0582E Wrong password.

Explanation: The existing keystore password that you provided did not match the actual password.

System action: The operation fails.

Administrator response: Specify the correct password value for the keystore. Then, try the operation again.

CTGKM0583E Group already exists: *VALUE_0*

Explanation: The group that you are attempting to create already exists.

System action: The operation fails.

Administrator response: Specify an alternative name for the group. Then, try the operation again.

CTGKM0584E The key in the certificate to be imported does not match the key in the original certificate request.

Explanation: The key of the certificate that returned from a Certificate Authority does not match the key in the original certificate request.

System action: The operation fails.

Administrator response: Import the certificate response using an alias that corresponds to this response. Then, try the operation again.

CTGKM0585E Error occurred while verifying the key and certificate.

Explanation: This is an internal database error.

System action: The operation fails.

Administrator response: The audit log might contain information about the error. Collect the information and contact IBM Support.

CTGKM0586E Keystore name and key alias must be specified.

Explanation: The keystore name or the key alias value is not correct. You might have attempted to list or delete a key and did not specify all required values, such as the keystore name.

System action: The operation fails.

Administrator response: Specify a valid keystore name and key alias. Alternatively, you might use the tklmKeyList command to determine which keys the keystore contains. Then, try the operation again.

CTGKM0587E Alias *VALUE_0* **does not exist in the keystore** *VALUE_1*

Explanation: The key alias is incorrectly specified. The alias does not exist in the specified keystore.

System action: The operation fails.

Administrator response: Ensure that the alias value is correct. Then, try the operation again.

CTGKM0588E Error occurred while loading data from the file *VALUE_0*

Explanation: This is an error in reading data from a key or certificate file.

System action: The operation fails.

Administrator response: Ensure that the path and filename are correct. Then, try the operation again.

CTGKM0589E Error occurred while retrieving the entry *VALUE_0* **from the keystore** *VALUE_1*

Explanation: Both the path name and the keystore password must be correctly specified.

System action: The operation fails.

Administrator response: Ensure that the keystore path name and password are valid. Then, try the operation again.

CTGKM0590E The file does not have any data.

Explanation: An attempt was made to import a private key from a private key file. The private key file is empty.

System action: The operation fails.

Administrator response: Ensure that you specified a file that contains the desired private key. Then, try the operation again.

CTGKM0591E *VALUE_0* **has more than one key entry; it is not supported by import operation.**

Explanation: You used the tklmKeyImport command to import a PKCS12 file with more entries than IBM Security Key Lifecycle Manager supports. Only one private key can be imported, using this command.

System action: The import key operation fails.

Administrator response: Import the key from a file that contains only one private key. Then, try the operation again.

CTGKM0592E *VALUE_0* **already exists.**

Explanation: The alias of a key that you attempted to import already exists in the keystore.

System action: The import key operation fails.

Administrator response: Specify a different alias. You might use the tklmKeyList command to view the keys that are currently in the keystore. Then, try the operation again.

CTGKM0593E **Entry in** *VALUE_0* **is not the private key entry.**

Explanation: You attempted to import a private key. However, the entry in the PKCS12 file is not a private key entry. It might be a certificate entry.

System action: The operation fails.

Administrator response: You might need to obtain a different file, and validate that the file has a private key. Then, try the operation again.

CTGKM0594E **No private key entry in the file** *VALUE_0*

Explanation: You attempted to import a private key. However, there is no private key entry in the PKCS12 file.

System action: The operation fails.

Administrator response: You might need to obtain a different file, and validate that the file has a private key. Then, try the operation again.

CTGKM0595E **Key alias cannot exceed 12 characters in length.**

Explanation: A key alias for a set of multiple keys has a prefix that must be 3 characters long. If you create only one key, the value for the alias cannot exceed 12 characters.

System action: The secret key or keys are not created.

Administrator response: Specify an alias that does not exceed the character limit. Then, try the operation again.

CTGKM0596E **Alias** *VALUE_0* **already exists in the keystore** *VALUE_1*

Explanation: You attempted to generate a key, but the key alias already exists in the keystore. A different alias is required.

System action: The secret key or keys are not created.

Administrator response: Specify a different alias. Then, try the operation again.

CTGKM0597E **Error occurred while generating the secret key.**

Explanation: The Java Cryptography Extension attempted to generate a secret key, but the process failed.

System action: The secret key or keys are not created.

Administrator response: Try the operation again. If the problem continues, collect any information that might be in the audit log and then contact IBM Support.

CTGKM0598E **The number of keys cannot be smaller than 1 or larger than 9999.**

Explanation: For the tklmSecretKeyCreate command, the value for the numOfKeys parameter must be a positive integer that is 1 or greater, and not larger than 9999.

System action: The secret keys are not created.

Administrator response: Specify a value for the number of keys that does not exceed the limit. Then, try the operation again.

CTGKM0599E **Error occurred while adding the key to the keystore.**

Explanation: The tklmSecretKeyCreate or the tklmKeyImport command experienced an error attempting to add a key to the keystore.

System action: The secret key or keys are not added to the keystore.

Administrator response: Collect any information that might be in the audit log and then contact IBM Support.

CTGKM0600E **Method Not Implemented:** *VALUE_0*

Explanation: This is an internal message that IBM Security Key Lifecycle Manager generates when a method call encounters an error condition.

System action: The operation fails.

Administrator response: This error should not occur in your environment. First, examine the audit log for exception information about the method call. You might need to contact IBM Support.

CTGKM0601E **An error occurred adding/updating value for attribute** *VALUE_0*

Explanation: An attempt was made to write a null value to the SKLMConfig.properties file. The file might be write protected. Alternatively, an internal component such as the key server component returned an error or was not available.

System action: The operation fails.

Administrator response: Ensure that the SKLMConfig.properties file is write enabled. If an internal component failed, you might restart the IBM Security Key Lifecycle Manager server. Then, try the operation again. If the problem continues, examine the audit log for exception information about the update to the attribute. Make necessary corrections. Then, try the operation again. You might need to call IBM Support.

CTGKM0602E An error occurred getting the value for attribute *VALUE_0*

Explanation: An attempt was made to read a value from the SKLMConfig.properties file. You might not be authorized to read the SKLMConfig.properties file.

System action: The operation fails.

Administrator response: Ensure that you have the correct authorization. If the problem continues, examine the audit log for exception information about the error. Make necessary corrections. Then, try the operation again. You might need to call IBM Support.

CTGKM0603E An error occurred deleting attribute *VALUE_0*

Explanation: An attempt was made to delete an attribute value from the SKLMConfig.properties file. The file might be write protected. Alternatively, an internal component such as the key server component returned an error or was not available.

System action: The operation fails.

Administrator response: Ensure that the SKLMConfig.properties file is write enabled. If an internal component failed, you might restart the IBM Security Key Lifecycle Manager server. Then, try the operation again. If the problem continues, examine the audit log for exception information about the update to the attribute. Make necessary corrections. Then, try the operation again. You might need to call IBM Support.

CTGKM0604E An error occurred merging configuration with file *VALUE_0*

Explanation: Merging in this context means that a problem occurred in merging two configuration files. This might occur during migration of configuration data from an existing Encryption Key Manager server. Data was not written from an existing configuration file into a new SKLMConfig.properties file. The new file might write protected, or you might not have sufficient authority.

System action: The operation fails.

Administrator response: Ensure that the SKLMConfig.properties file is write enabled and that you have appropriate access to the file. You might also examine the audit log for exception information. Make necessary corrections. Then, try the operation again. You might need to contact IBM Support.

CTGKM0605E An error occurred replacing configuration file with file *VALUE_0*

Explanation: Configuration file replacement might occur during migration of configuration data from an existing Encryption Key Manager server. You might not have sufficient authority to replace the configuration file.

System action: The operation fails.

Administrator response: Ensure that you have appropriate access to the file. You might also examine the audit log for exception information. Make necessary corrections. Then, try the operation again. You might need to contact IBM Support.

CTGKM0615E Keystore password cannot exceed 175 single-byte or 87 double-byte characters.

Explanation: Keystore password cannot exceed 175 single-byte or 87 double-byte characters.

System action: Keystore password update fails.

Administrator response: Enter a password not greater than 175 single-byte or 87 double-byte characters.

CTGKM0616E Device with UUID *VALUE_0* **does not belong to the device group** *VALUE_1***.**

Explanation: The device with the specified UUID does not belong to the specified device group.

System action: Device list fails.

Administrator response: Enter the correct device group, or leave it blank.

CTGKM0617E authorization.provider.class cannot be a numeric string.

Explanation: authorization.provider.class cannot be a numeric string.

System action: The operation fails.

Administrator response: Specify a non-numeric value.

CTGKM0618E *VALUE_0* **can only be positive integer.**

Explanation: The value must be an integer greater than zero.

System action: The operation fails.

Administrator response: Specify a positive integer.

CTGKM0620E Error parsing XML template.

Explanation: This is an internal message that IBM Security Key Lifecycle Manager generates after problems occur reading an XML template file that has syntax errors.

System action: The operation fails.

Administrator response: This error should not occur in your environment. First, examine the audit log for exception information about the XML parsing error. You might need to contact IBM Support.

CTGKM0621E Error in XML element *VALUE_0* **, expecting** *VALUE_1*

Explanation: This is an internal message that IBM Security Key Lifecycle Manager generates after problems occur reading an XML template file that has errors.

System action: The operation fails.

Administrator response: This error should not occur in your environment. First, examine the audit log for exception information about the error. You might need to contact IBM Support.

CTGKM0622E XML attribute *VALUE_0* **missing for element** *VALUE_1*

Explanation: This is an internal message that IBM Security Key Lifecycle Manager generates after problems occur reading an XML template file that has errors.

System action: The operation fails.

Administrator response: This error should not occur in your environment. First, examine the audit log for exception information about the error. You might need to contact IBM Support.

CTGKM0623E Error in XML template attribute value *VALUE_0* **for attribute** *VALUE_1*

Explanation: This is an internal message that IBM Security Key Lifecycle Manager generates after problems occur reading an XML template file that has errors.

System action: The operation fails.

Administrator response: This error should not occur in your environment. First, examine the audit log for exception information about the error. You might need to contact IBM Support.

CTGKM0624E Template *VALUE_0* **not found.**

Explanation: This is an internal message that IBM Security Key Lifecycle Manager generates after problems occur attempting to read an XML template file.

System action: The operation fails.

Administrator response: This error should not occur in your environment. First, examine the audit log for exception information about the error. You might need to contact IBM Support.

CTGKM0630E Validation error: *VALUE_1* **for parameter** *VALUE_0***.**

Explanation: When a command is parsed, this error occurs if you enter a parameter value that is not valid.

System action: The operation fails.

Administrator response: Specify a valid value for the parameter. Then, try the operation again.

CTGKM0631E Missing required parameter *VALUE_0* **.**

Explanation: The value that you specified for the required parameter is blank or missing.

System action: The operation fails.

Administrator response: Specify a valid value for the parameter. Then, try the operation again.

CTGKM0632E Missing required *VALUE_0* **parameter** *VALUE_1*

Explanation: The required parameter value is blank or missing.

System action: The operation fails.

Administrator response: Specify a valid value for the required parameter. Then, try the operation again.

CTGKM0633E Validation error: *VALUE_1* **is not supported for parameter** *VALUE_0* **. Valid values are:** *VALUE_2*

Explanation: The value that you specified for the parameter is not valid.

System action: The operation fails.

Administrator response: Ensure that the value that you specified is valid. Then, try the operation again.

CTGKM0634E Validation Error: *VALUE_1* **for parameter** *VALUE_0* **, must be 16 characters long and contain valid characters.**

Explanation: When a command is parsed, this error occurs if you enter a parameter value that is not valid.

System action: The operation fails.

Administrator response: Specify a valid value for the parameter. Then, try the operation again.

CTGKM0635E Incorrect syntax for required parameter attribute. Syntax is {attribute1}{attribute2}..{attributeN}.

Explanation: The syntax for the parameter 'attribute' was incorrect.

System action: The operation fails.

Administrator response: Correct the syntax, and try again.

CTGKM0640E Certificate Request file already exists: *VALUE_0*

Explanation: The file name that you specified in the certificate request matches a certificate request file name that currently exists.

System action: The operation fails.

Administrator response: Specify a different file name for the certificate request. For example, specify myUniqueRequest.crt. Then, try the operation again.

CTGKM0641E Error in writing file *VALUE_0* **:** *VALUE_1*

Explanation: You might not have authorization to write a certificate request file, or the path name that you specified is incorrect.

System action: The operation fails.

Administrator response: Ensure that you have appropriate access to the file, and that the path and file names are correctly specified. Then, try the operation again. You might also examine the audit log for exception information about the file operation. Make necessary corrections. Then, try the operation again. You might need to contact IBM Support.

CTGKM0645E Device *VALUE_0* **not found.**

Explanation: An attempt to read device information used an identifier for the device does not match an existing device serial number. This is an internal error.

System action: The operation fails.

Administrator response: Determine whether the value that you specified matches an existing device, and that the specified type matches the device group. For example, you might use the tklmDeviceList command to identify existing devices of a given group. Correct the device specification. Then, try the operation again.

CTGKM0650E Error while attempting to encrypt a file using algorithm: *error_msg*

Explanation: Error occurred while attempting to encrypt a file using algorithm

System action: Operation invoking the encryption fails.

Administrator response: Error can occur for a number of reasons: missing algorithm, incorrect algorithm parameter, incorrect encryption key or key specification (password), incorrect padding mechanism. This error is not expected to occur. If it does, the IBM Security Key Lifecycle Manager administrator should investigate the cause.

CTGKM0651E Error while attempting to decrypt a file using algorithm: *error_msg*

Explanation: This error occurred while attempting to decrypt a file using the algorithm.

System action: The operation fails.

Administrator response: This error can occur for a number of reasons: missing algorithm, incorrect algorithm parameter, incorrect encryption key or key specification (password), or an incorrect padding mechanism. This error is not expected to occur. If it does, the IBM Security Key Lifecycle Manager administrator should investigate the cause.

CTGKM0660E tklmCertList supports the optional parameters -usage and -v, and these parameter combinations: No parameters; or -uuid; or -alias and -keystoreName; or -attributes.

Explanation: The command failed because the combination of parameter values that you specified is not valid.

System action: The operation fails.

Administrator response: Specify no parameter, or specify a value for alias, or specify an attribute, or specify both a uuid and keystoreName. Then, try the operation again.

CTGKM0704E Group name cannot contain \' \\ \'.

Explanation: The group name must not contain the characters \' \\ \' in the name.

System action: The operation fails.

Administrator response: Specify a group name that does not contain \' \\ \'. Then, try the operation again.

CTGKM0705E Either group name or group UUID must be specified.

Explanation: The command requires that you specify either the group name or the group UUID.

System action: The operation fails.

Administrator response: Specify either a group name or group UUID. Then, try the operation again.

CTGKM0706E The key group with the name *VALUE_0* **does not have the UUID** *VALUE_1***.**

Explanation: The specified group name and group UUID do not match.

System action: The operation fails.

Administrator response: Correct the group name or

UUID, or specify only one of the two. Then, try the operation again.

CTGKM0742E Key type not valid: ***VALUE_0*.**

Explanation: Key type entered is not valid.

System action: Key operation fails.

Administrator response: Check the logs for more information.

CTGKM0750E Validation Error: ***VALUE_0*** **for compromised, only y is allowed.**

Explanation: When you use the tklmCertUpdate command to specify that a certificate is compromised, the only valid value is y (compromised). You cannot change a compromised certificate to an uncompromised state.

System action: The certificate state is not changed.

Administrator response: Specify a value of y for the -compromised attribute. Then, try the operation again.

CTGKM0751E Conflicting parameter values specified for compromised and trusted.

Explanation: When you use the tklmCertUpdate command to specify that a certificate is compromised, the certificate cannot be marked as trusted. A certificate can only be marked trusted if it is not in expired, retired or compromised state.

System action: The certificate state is not changed.

Administrator response: Do not specify a value of y for trusted if the certificate needs to be marked as compromised.

CTGKM0752E Validtion Error: ***VALUE_0*** **for trusted, only y or n is allowed.**

Explanation: When you use the tklmCertUpdate command to specify a certificate is trusted or not, the only valid values are y (trusted) or n (not trusted).

System action: The certificate attribute is not changed.

Administrator response: Specify a value of y or n for the -trusted attribute. Then, try the operation again.

CTGKM0753E Certificate is not in a valid state to mark it trusted.

Explanation: If the certificate is in compromised, destroyed, expired or retired state then it cannot be marked as trusted. Only certificates in an active or pre-active state can be marked as trusted.

System action: The certificate attribute is not changed.

Administrator response: Check the state of the certificate using the tklmCertList command. Do not specify trusted attribute to mark the certificate as trusted if the certificate state is compromised, destroyed, expired or retired.

CTGKM0758E Certificate with alias ***VALUE_0*** **cannot be deleted because this certificate is specified as an SSL/KMIP or IKEv2-SCSI certificate and is in use.**

Explanation: Using the tklmCertDelete command, you cannot delete a certificate that is specified as the SSL/KMIP or IKEv2-SCSI certificate.

System action: The certificate delete operation fails.

Administrator response: Specify a different certificate as the SSL/KMIP or IKEv2-SCSI certificate. Ensure that the certificate that you intend to delete no longer has the specification. Then, try the operation again.

CTGKM0759E Key with alias ***VALUE_0*** **cannot be deleted because the certificate associated with this key is specified as an SSL/KMIP or IKEv2-SCSI certificate and is in use.**

Explanation: Using the tklmKeyDelete command, you cannot delete a key for which certificate is specified as the SSL/KMIP or IKEv2-SCSI certificate.

System action: The key delete operation fails.

Administrator response: Specify a different certificate as the SSL/KMIP or IKEv2-SCSI certificate. Ensure that the key that you intend to delete no longer has the specification. Then, try the operation again.

CTGKM0760E Certificate with alias = ***VALUE_0*** **cannot be deleted because this certificate is specified as a default or partner certificate.**

Explanation: Using the tklmCertDelete command, you cannot delete a certificate that is specified as the system default or partner certificate.

System action: The certificate delete operation fails.

Administrator response: Specify a different certificate as the system default or partner certificate. Ensure that the certificate that you intend to delete no longer has the specification. Then, try the operation again.

CTGKM0761E Key with alias = ***VALUE_0*** **cannot be deleted because the certificate associated with this key is specified as a default or partner certificate.**

Explanation: Using the tklmKeyDelete command, you cannot delete a key which certificate is specified as the system default or partner.

System action: The key delete operation fails.

Administrator response: Specify a different certificate as the system default or partner. Ensure that the key that you intend to delete no longer has the specification. Then, try the operation again.

CTGKM0775E Command is not supported in FIPS mode.

Explanation: Algorithms used by this command are not supported in a FIPS mode environment.

System action: Command Fails

Administrator response: Change the value of the FIPS property to off and restart the server before using this command.

CTGKM0776W The number of device audit entries returned reaches the limit of 2000 records. Only the first 2000 entries are displayed. You might need to specify a different filter for your search.

Explanation: The list operation only fetches the first 2000 rows.

System action: The first 2000 entries are displayed.

Administrator response: The result set reached the 2000 entries limit. Specify a different filter, and try the operation again.

CTGKM0800E Attempt to insert entry with preexisting primary key value failed on table *VALUE_0*

Explanation: Using the tklmDeviceAdd command, a primary key in the database uniquely represents a device with a combination of several parameters (serialNumber, type, and worldwideName). The primary key already exists. The device that you attempted to add already exists in the database.

System action: The device is not added to the database.

Administrator response: Specify different values for the device that you intend to add. Then, try the operation again.

CTGKM0801E A key group name and a key alias cannot be same in the IBM Security Key Lifecycle Manager database. Value that you specified: *VALUE_0*

Explanation: Before creating metadata for a key, IBM Security Key Lifecycle Manager verifies that a key group does not already exist with the same name as the alias. Similarly, before creating a key group, the IBM Security Key Lifecycle Manager verifies that a key does not exist with the same alias as the group name.

System action: The create operation fails.

Administrator response: If you are creating a key, specify a different alias name. If you are creating a key group, specify a different key group name.

CTGKM0802E The backup program could not determine the database name from the data source URL: *VALUE_0*

Explanation: The IBM Security Key Lifecycle Manager determines the name of database to backup by parsing the data source URL that was specified during the installation of the IBM Security Key Lifecycle Manager database. The backup program failed because the database name could not be determined.

System action: The database backup operation fails.

Administrator response: Verify that the the data source URL is correctly specified. Then, try the operation again.

CTGKM0803E The backup program could not determine the directory to which database files will be saved.

Explanation: IBM Security Key Lifecycle Manager determines the directory to save the files by using the datastore.properties file. The backup program failed because the directory could not be determined.

System action: The database backup operation fails.

Administrator response: Verify that the IBM Security Key Lifecycle manager has been configured correctly and the correct value for the tklm.backup.db2.dir property exists in the properties file.

CTGKM0804E The backup program failed because archival logging has not been enabled.

Explanation: The IBM Security Key Lifecycle Manager requires that archival logging is enabled for the IBM Security Key Lifecycle Manager in order to perform an online backup. The installation program enables the archival logging.

System action: The database backup operation fails.

Administrator response: Verify that the IBM Security Key Lifecycle Manager has been configured correctly and that archival logging is enabled. To determine the logging method that DB2 currently uses, run this command: db2 get db cfg for sklmdb

CTGKM0805E The backup program failed because the backup program could not write to the directory that was specified for the database files.

Explanation: IBM Security Key Lifecycle Manager requires that directory that is specified in the datastore.properties file exists for the property tklm.backup.db2.dir and is writeable. The DB2 backup utility found an error in the specified directory.

System action: The database backup operation fails.

Administrator response: Verify that the directory specified in the properties file exists and is writeable.

CTGKM0806E An error occurred reading the sklm.properties file.

Explanation: IBM Security Key Lifecycle Manager reads the sklm.properties file to determine appropriate parameters to perform a database restore operation. An error occurred while reading this file.

System action: The database restore operation fails.

Administrator response: Ensure that the sklm.properties file is read enabled. If an internal component failed, you might restart the IBM Security Key Lifecycle Manager server. Then, try the operation again. If the problem continues, examine the audit log for exception information. Make necessary corrections. Then, try the operation again. You might need to contact IBM Support.

CTGKM0807E An error occurred getting the value for attribute *VALUE_0*

Explanation: An attempt was made to read a value from the sklm.properties file. You might not be authorized to read the sklm.properties file.

System action: The database restore operation fails.

Administrator response: Ensure that you have the correct authorization. If the problem continues, examine the audit log for exception information about the error. Make necessary corrections. Then, try the operation again. You might need to call IBM Support.

CTGKM0808E The restore program could not determine the directory from which to restore database files.

Explanation: IBM Security Key Lifecycle Manager determines the directory to restore the file from arguments specified to the restore program. If the arguments are not specified, it determines the directory from the datastore.properties file. The restore program failed because the directory could not be determined.

System action: The database restore operation fails.

Administrator response: Verify that IBM Security Key Lifecycle Manager has been configured correctly and a valid tklm.backup.db2.dir value exists in the datastore.properties file.

CTGKM0809E A null value was specified for the timestamp associated with the previously saved database files.

Explanation: IBM Security Key Lifecycle Manager determines the timestamp to restore the file from arguments specified to the restore program. The restore program failed because the timestamp could not be determined.

System action: The database restore operation fails.

Administrator response: Verify that IBM Security Key Lifecycle manager has been configured correctly.

CTGKM0850E An exception occurred during the restore operation. Examine the db2restore.log for exception information. Complete the restore operation before attempting any other IBM Security Key Lifecycle Manager tasks.

Explanation: IBM Security Key Lifecycle Manager encountered an exception.

System action: The database restore operation fails.

Administrator response: Ensure that you have configured IBM Security Key Lifecycle Manager correctly. If the problem continues, examine the db2tklmrestore log for exception information about the error. Make necessary corrections. Then, try the operation again. You might need to contact IBM Support.

CTGKM0851E The group cannot be created because an entity (key) cannot be in multiple key groups. The entity *VALUE_0* **is already a member of the group** *VALUE_1***.**

Explanation: While creating a group, you specified an entity that already exists in another group. The entity cannot not be in multiple groups.

System action: The group creation operation fails.

Administrator response: Specify an entity (key) that does not exist in another group. Then, try the operation again.

CTGKM0852E The specified entity (key) *VALUE_0* **cannot be added to the group because the entity cannot be in multiple groups. The entity is already a member of the group** *VALUE_1***.**

Explanation: While adding an entity to a group, you specified an entity that already exists in another group. The entity cannot not be in multiple groups.

System action: The group creation operation fails.

Administrator response: Specify an entity (key) that does not exist in another group. Then, try the operation again.

CTGKM0900E Database connection failed on data source *VALUE_0*. Check to see if database server needs to be restarted or if database user's password needs to be updated.

Explanation: The database may be down or the database user's password for accessing the IBM Security Key Lifecycle Manager database might need to be updated.

System action: The database operation fails.

Administrator response: Verify that the database is up or check if the database user's password for accessing the IBM Security Key Lifecycle Manager database might need to be updated, for example, after a mandated password change on the corresponding operating system account.

CTGKM0901E Backup directory could not be resolved. Try specifying a valid backup directory explicitly.

Explanation: Backup directory could not be resolved.

System action: The IBM Security Key Lifecycle Manager backup operation fails.

Administrator response: Make sure that the backup directory provided to the backup operation is valid. Try specifying a valid backup directory explicitly. If this is the first time you are executing backup, find a directory suitable for IBM Security Key Lifecycle Manager backup and specify the directory explicitly on the backup command.

CTGKM0902E Default backup directory cannot be determined: *VALUE_0* **.**

Explanation: The configuration property tklm.backup.dir cannot be found.

System action: The IBM Security Key Lifecycle Manager backup operation fails.

Administrator response: Ensure that the tklm.backup.dir parameter is set in the configuration file or provide the backupDirectory parameter to the backup operation.

CTGKM0903W IBM Security Key Lifecycle Manager restore is already in progress.

Explanation: Only one restore/backup operation is allowed at any given time.

System action: The requested operation will not be performed.

Administrator response: Wait until the operation completes.

CTGKM0904W IBM Security Key Lifecycle Manager backup is already in progress.

Explanation: Only one restore/backup operation is allowed at any given time.

System action: The requested operation will not be performed.

Administrator response: Wait until the operation completes.

CTGKM0905E Backup of *VALUE_0* **failed:** *VALUE_1* **.**

Explanation: Backup failed unexpectedly.

System action: Backup fails.

Administrator response: Check the log entries to find out the reason for backup failure.

CTGKM0906E Specified backup file does not exist: *VALUE_0* **.**

Explanation: The backup file does not exist.

System action: The IBM Security Key Lifecycle Manager restore operation fails.

Administrator response: Make sure that the backup file path provided to the restore operation is valid.

CTGKM0907E Restore of *VALUE_0* **failed:** *VALUE_1* **.**

Explanation: Restore failed unexpectedly.

System action: Restore fails.

Administrator response: Check the log entries to find out the reason for restore failure.

CTGKM0908E Error reading manifest from the backup jar file: *VALUE_0* **.**

Explanation: Backup manifest could be corrupted or incorrect for unknown reasons.

System action: Restore fails.

Administrator response: Check the log entries to find out the reason for restore failure.

CTGKM0909E Error making a backup copy of existing configuration file: *VALUE_0* **.**

Explanation: Backup copy of a configuration file could not be made.

System action: Restore fails.

Administrator response: Check the log entries to find out the reason for restore failure.

CTGKM0910E I/O error while creating backup jar file *jar_file_name* **\nError message:** *error_msg* **.**

Explanation: Input/Output file error occurred while creating the backup jar file.

System action: Backup fails.

Administrator response: Check the log entries to find out the reason for backup failure.

CTGKM0911E Entry in the jar has been modified since the backup file was created: \nJar file: *jar_file_name* **\nJar entry name:** *jar_entry_name* **\nDestination:** *destination*

Explanation: An entry in the backup jar file has been changed since the jar file was created. This may indicate a corrupt file or a backup jar which has been tampered with in some way.

System action: Restore fails.

Administrator response: Check the integrity of the backup file in question. Use another backup jar file if available.

CTGKM0912E I/O error, missing or corrupt data detected while extracting jar file entry.\nJar file name: *jar_file_name* **\nJar entry name:** *jar_entry_name* **\nDestination:** *destination*

Explanation: Backup jar file or an entry in the backup jar file is missing, corrupt or has been tampered with.

System action: Restore fails.

Administrator response: Check the integrity of the backup file in question. Use another backup jar file if available.

CTGKM0913E I/O error while decrypting and/or extracting jar file entry.\nJar file name: *jar_file_name* **\nJar entry name:** *jar_entry_name* **\nDestination:** *destination* **\nError message:** *error_msg* **\nPossible cause: Incorrect password may have been provided, which would result in a jar entry indicated above being decrypted incorrectly.**

Explanation: I/O error occurred while decrypting and/or extracting a jar file entry. A common reason for this error is that a file was encrypted using one password and an incorrect or null password was provided to decrypt the file. Decryption will still take place, but the decrypted file is not valid and cannot be used in later stages of the restore.

System action: Restore fails.

Administrator response: Check the password provided for decryption. Check if the destination location contains sufficient disk space to hold the decrypted/extracted file. Check the log files for additional clues if necessary. Retry the operation.

CTGKM0914W IBM Security Key Lifecycle Manager backup or restore operation is in progress. Database connections cannot be obtained at this moment. Try again later.

Explanation: To minimize the risk of data inconsistency, a IBM Security Key Lifecycle Manager backup or restore operation needs to be done while the database is not used by the IBM Security Key Lifecycle Manager server. This warning message will be displayed in the graphical user interface and/or log file.

System action: Database connection is not possible and the IBM Security Key Lifecycle Manager operation requesting it fails.

Administrator response: Retry your operation after the backup or restore is done.

CTGKM0915E IBM Security Key Lifecycle Manager backup failed due to low disk space on the file system containing the backup directory *VALUE_0***.**

Explanation: The database backup failed due to low disk space on the file system containing the backup directory.

System action: The backup operation fails.

Administrator response: Increase the available disk space on the file system containing the database backup directory. Then try the operation again.

CTGKM0916W Warning: Information about this backup was not saved. However, the IBM Security Key Lifecycle Manager backup operation saved all the necessary files, which could be used for a restore operation in the future.

Explanation: A successful backup operation saves information about the backup in a file named SKLM_HOME/config/lastbackupinfo. However, the step to save this information failed, or the information is incorrect. You can still use the backup file for a restore operation. Possible reasons for failure: Another process was using the lastbackupinfo file, or permissions for the SKLM_HOME/config directory are incorrect.

System action: The backup operation succeeded but the information about the most recent backup could not be saved.

Administrator response: Verify that no other process is using the lastbackupinfo file and the SKLM_HOME/config directory has the correct read and write permissions. Then try the backup operation again if you want this information to be accurate.

CTGKM0917E IBM Security Key Lifecycle Manager failed to determine the information about the last successful backup. Possible reasons: A backup was never done on this system, the file permissions are incorrect, or the file was deleted.

Explanation: After a successful backup, IBM Security Key Lifecycle Manager saves the information about the backup in a file named SKLM_HOME/config/lastbackupinfo. This file could not be read.

System action: The IBM Security Key Lifecycle Manager fails to determine the information about the last successful backup.

Administrator response: If you restored a backup from another system, this file may not be available. Verify that the SKLM_HOME/config/lastbackupinfo file exists and that the file has read permission. Run the operation again.

CTGKM0918E Cannot add a value to a single-valued KMIP attribute when a value already exists. Object uuid: *VALUE_0*, **Object name:** *VALUE_1*, **Attribute name:** *VALUE_2*.

Explanation: Cannot add a value to a single-valued KMIP attribute when value already exists. Existing value can be modified or deleted.

System action: The operation fails.

Administrator response: Modify or delete the value. Then, try the operation again.

CTGKM0919E Cannot modify attribute value. Current value is not defined at index *VALUE_0*. **Object uuid:** *VALUE_1*, **Object name:** *VALUE_2*, **Attribute name:** *VALUE_3*.

Explanation: Cannot modify a value on a multi-valued KMIP attribute because the value does not exist at the specified index.

System action: The operation fails.

Administrator response: Modify or delete the value. Then, try the operation again.

CTGKM0920E Cannot delete attribute value. Incorrect attribute value index was specified: *VALUE_0*. **Object uuid:** *VALUE_1*, **Object name:** *VALUE_2*, **Attribute name:** *VALUE_3*.

Explanation: Cannot delete a value because the value does not exist at the specified index.

System action: The operation fails.

Administrator response: Provide a valid index.

CTGKM0921E Cannot modify attribute value. New value is not provided. Object uuid: *VALUE_0*, **Object name:** *VALUE_1*, **Attribute name:** *VALUE_2*.

Explanation: Cannot modify a value because a new value was not provided.

System action: The operation fails.

Administrator response: Provide a valid new value or delete the existing value.

CTGKM0922E Range Specification is not valid: *VALUE_0*

Explanation: One or two range-capable values of the same range-capable attribute type are expected.

System action: The operation fails.

Administrator response: Provide a valid range. Then, try the operation again.

CTGKM0923E Unsupported parameter type detected: *VALUE_0*

Explanation: Unsupported parameter type was specified.

System action: The operation fails.

Administrator response: Provide a valid parameter. Then, try the operation again.

CTGKM0924E Unsupported usage mask '*VALUE_0***' on object with uuid** *VALUE_1*

Explanation: Unsupported KMIP usage mask was detected in the data store.

System action: The IBM Security Key Lifecycle Manager cannot retrieve object's data.

Administrator response: Investigate the source of incorrect data and correct or delete it before trying again.

CTGKM0925E Cannot *VALUE_0* **a value when no attribute value was supplied. Object uuid:** *VALUE_1*, **Object name:** *VALUE_2*, **Attribute name:** *VALUE_3*.

Explanation: Cannot perform the operation when no attribute value is passed.

System action: The operation fails.

Administrator response: Supply the value. Then, try the operation again.

CTGKM0926E Cannot *VALUE_0* **a value when no attribute value currently exists. Object uuid:** *VALUE_1*, **Object name:** *VALUE_2*, **Attribute name:** *VALUE_3*.

Explanation: Cannot perform the operation when that attribute does not exist on that object.

System action: The operation fails.

Administrator response: Add the attribute with its value.

CTGKM0927E Cannot *VALUE_0* **a value when no attribute value currently exists at that index. Object uuid:** *VALUE_1*, **Object name:** *VALUE_2*, **Attribute name:** *VALUE_3*. **Index:** *VALUE_4*.

Explanation: Cannot perform the operation when that attribute does not exist on that object.

System action: The operation fails.

Administrator response: Add the attribute with its value.

CTGKM0928E Certificate alias cannot contain \' \\ \' or \' / \' or \' .

Explanation: You cannot create a certificate with \' \\ \' or \' / \' or \' in the alias.

System action: The operation fails.

Administrator response: Specify an alias that does not contain \' \\ \' or \' / \' or \' . Then, try the operation again.

CTGKM0929E Error occurred while looking up version numbers.

Explanation: Command could not be executed. On Windows systems, ensure that the environment variable ProgramFiles is set correctly. If you are running 64-bit Windows, ensure that the environment variable ProgramFiles(x86) is also correctly set.

System action: The operation fails.

Administrator response: Ensure that the required environment variables are correctly set.

CTGKM0930E Because this certificate is outside of its validity period and the validate certificates check is enabled, cannot assign this certificate to the device.

Explanation: The certificate is outside of its validity period and not available for use.

System action: The operation fails.

Administrator response: Select another certificate.

CTGKM0931E Not all version numbers were successfully retrieved.

Explanation: Some version numbers could not be retrieved. Ensure that the 'IBM ADE Service' service is running.

System action: The operation fails.

Administrator response: Ensure that the service 'IBM ADE Service' is running. To start the service in Windows, go to the Control Panel > Services. To start the service in Linux, run the /usr/ibm/common/acsi/bin/acsisrv.sh command. Then, try the operation again.

CTGKM0932E The value of newAlias cannot be an empty string.

Explanation: You must provide a value for newAlias.

System action: The operation fails.

Administrator response: Specify a value for newAlias. Then, try the operation again.

CTGKM0933E Certificate with alias *VALUE_0* **is compromised and cannot be set as a system, partner, or device default.**

Explanation: A compromised certificate cannot be set as a system, partner, or device default.

System action: The operation fails.

Administrator response: Select another certificate to use as a default.

CTGKM0934E Key with alias *VALUE_0* **is compromised and cannot be set as a device default.**

Explanation: A compromised key cannot be set as a device default.

System action: The operation fails.

Administrator response: Select another key to use as a default.

CTGKM0935E alias is required if newAlias is specified.

Explanation: The alias parameter is required if newAlias is specified.

System action: The operation fails.

Administrator response: Specify an alias and try the operation again.

CTGKM0937E Target uuid is not valid for the rollover object to be deleted: *VALUE_0*

Explanation: The target uuid is not valid for the rollover object.

System action: The operation fails.

Administrator response: Ensure that the target uuid is valid and try the operation again.

CTGKM0938E Device group of the target is not valid for the rollover object to be deleted: *VALUE_0*

Explanation: The target device group is not valid for the rollover object.

System action: The operation fails.

Administrator response: Ensure that the target device group is valid and try the operation again.

CTGKM0940E Common name cannot exceed 256 characters.

Explanation: Common name is too long. The maximum length allowed is 256 characters.

System action: Certificate creation fails.

Administrator response: Specify a common name not exceeding 256 characters, then try the operation again.

CTGKM0942E Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files are required. For more information, see the "Backup and restore" section of IBM Security Key Lifecycle Manager documentation on IBM Knowledge Center.

Explanation: For Stronger encryption we require Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files. For more information, see the "Backup and restore" section of IBM Security Key Lifecycle Manager documentation on IBM Knowledge Center.

Administrator response: Install the Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files.

CTGKM1000E Error while invoking scheduled task handler:\n \tTask Id: *task_id*\n **\tTask Type:** *task_type*\n **\tTask Name:** *task_name*\n **\tOriginal error message:** *error_message*

Explanation: Error occurred while instantiating a scheduled task handler class.

System action: Task will not be executed. Audit log will contain a failure message.

Administrator response: This should not happen under normal conditions. Make sure that the task handler class exists in the correct location and is specified correctly.

CTGKM1001E Required task name value is not defined.

Explanation: Required task name value was detected missing when scheduling a task.

System action: Task will not be scheduled.

Administrator response: This should not happen under normal conditions. Make sure that the task information contains all required values before submitting it.

CTGKM1002E Required task type value is not defined.

Explanation: Required task type value was detected missing when scheduling a task.

System action: Task will not be scheduled.

Administrator response: This should not happen under normal conditions. Make sure that the task information contains all required values before submitting it.

CTGKM1003E Error while scheduling a task:\n Original error message: *error_message*

Explanation: Required task type value was detected missing when scheduling a task.

System action: Task will not be scheduled.

Administrator response: This should not happen under normal conditions. Make sure that the task information contains all required values before submitting it.

CTGKM1004E Error occurred while suspending a scheduled task:\n Original error message: *error_message*

Explanation: Error occurred while suspending a scheduled task.

System action: Task will not be suspended.

Administrator response: This should not happen under normal conditions. Investigate the logs for clues.

CTGKM1005E Error occurred while resuming a suspended scheduled task:\n Original error message: *error_message*

Explanation: Error occurred while resuming a suspended scheduled task.

System action: Task will not be resumed.

Administrator response: This should not happen under normal conditions. Investigate the logs for clues.

CTGKM1006E Error occurred while canceling a scheduled task:\n Original error message: *error_message*

Explanation: Error occurred while canceling a scheduled task.

System action: Task will not be canceled.

Administrator response: This should not happen under normal conditions. Investigate the logs for clues.

CTGKM1007E Error occurred while purging a task:\n Original error message: *error_message*

Explanation: Error occurred while purging a task.

System action: Task will not be purged.

Administrator response: This should not happen under normal conditions. Investigate the logs for clues.

CTGKM1008E Scheduler is not available:\n Original error message: *error_message*

Explanation: Scheduler is not available.

System action: Scheduler related operations fail.

Administrator response: This should not happen under normal conditions. Investigate the logs and server startup log for clues. Restart the system. Check if the database server is operating correctly.

CTGKM1009E Error in scheduler task detected:\n Original error message: *error_message*

Explanation: Error in scheduler task detected.

System action: Operations involving this task will fail.

Administrator response: This should not happen under normal conditions. Investigate the logs for clues. Delete the task and reschedule.

CTGKM1015E Key server is down.

Explanation: The key server is an internal component that the IBM Security Key Lifecycle Manager server contains. There might be a protocol or a certificate error or the database might not be started when the IBM Security Key Lifecycle Manager server comes up.

System action: Keys are not served.

Administrator response: You might need to correct a protocol or certificate specification or start the database. Examine the audit log for error messages. After making corrections, restart the IBM Security Key Lifecycle Manager server. You might need to contact IBM Support.

CTGKM1016E TCP port not available.

Explanation: The TCP port did not initialize. Security Key Lifecycle Manager is not ready to accept key requests on the TCP port. It might be that another process has the port or that the socket timed out.

System action: The TCP port fails to initialize and TCP communication fails.

Administrator response: Ensure that no other program is using the TCP port. If the other process must have the port, specify a new TCP port number. Then restart the IBM Security Key Lifecycle Manager server. If the problem continues, examine the audit log. After making corrections, restart the IBM Security Key Lifecycle Manager server. You might need to contact IBM Support.

CTGKM1017E SSL failed to initialize.

Explanation: The SSL port did not initialize. Security Key Lifecycle Manager is not ready to accept key requests from a client on the SSL port. It might be that another process has the port or that the socket timed out.

System action: The SSL port fails to initialize and SSL communication fails.

Administrator response: Ensure that no other program is using the SSL port. If the other process must have the port, specify a new SSL port number. Then restart the IBM Security Key Lifecycle Manager server. If the problem continues, examine the audit log. After making corrections, restart the IBM Security Key Lifecycle Manager server. You might need to contact IBM Support.

CTGKM1018E KMIP failed to initialize.

Explanation: An internal error occurred. IBM Security Key Lifecycle Manager could not complete initialization for KMIP. It might be that another process has the port or that the socket timed out.

System action: The KMIP SSL port fails to initialize and IBM Security Key Lifecycle Manager cannot accept KMIP requests from a client.

Administrator response: Ensure that no other program is using the KMIP SSL port. If the other process must have the port, specify a new KMIP SSL port number. Then restart the IBM Security Key Lifecycle Manager server. If the problem continues, examine the audit log. After making corrections, restart the IBM Security Key Lifecycle Manager server. You might need to contact IBM Support.

CTGKM1020E Each port value must be unique. Two or more of the port values configured have the same value.

Explanation: TCP, SSL, and KMIP SSL cannot use the same port.

System action: The TCP, SSL, or KMIP SSL port values will not be saved until the values are updated to be unique values.

Administrator response: Set the TCP, SSL, and KMIP SSL ports to unique values.

CTGKM1050E An entry in a required field is missing, or an entry is not valid. The field is highlighted below.

Explanation: You attempted to change to another page or to submit changes when either a required field is not complete or you entered a value that is not valid.

System action: Additional page help appears by the field that has the error.

Administrator response: Correct the entry. Then submit the change again.

CTGKM1051E Because this certificate is currently the ${0} certificate and is in use, it cannot be deleted. If you still wish to delete this certificate, you must change the ${0} certificate.

Explanation: You cannot delete a certificate that is currently specified as the default or the system partner certificate, or a certificate that is currently assigned as the in-use SSL/KMIP or IKEv2-SCSI server certificate.

System action: The operation fails.

Administrator response: Specify another certificate as the current default or system partner certificate, or the in-use SSL/KMIP or IKEv2-SCSI server certificate. Then try to delete this certificate again.

CTGKM1052W Certificate will be deleted only if it is not currently used by any device. Any ${0} written previously using this certificate will be non-readable if the certificate is deleted. Are you sure you would like to try to delete ${1}?

Explanation: You cannot delete a certificate that is currently used by a device. Delete certificates only when the data protected by those certificates is no longer needed.

System action: If you proceed, the certificate is deleted. Deleting a certificate marks the certificate as destroyed in the database and deletes the material from the keystore.

Administrator response: Continue to delete the certificate, assuming it is not in use by a device.

CTGKM1053W Are you sure you would like to delete the device with device serial number ${0}?

Explanation: This message confirms that you want to delete the selected device.

System action: Confirming the message deletes the device.

Administrator response: Ensure that the serial number correctly identifies the device that you intend to delete. Then click **OK** to delete the device.

CTGKM1054E Because this key group is currently the ${0} key group and is in use, it cannot be deleted. If you still wish to delete this key group, you must change the ${0} key group.

Explanation: IBM Security Key Lifecycle Manager could not delete a key group that is a default.

System action: The key group is retained in the IBM Security Key Lifecycle Manager database.

Administrator response: Ensure that the key group is not the default or reassign another key group to be the system default. Then try the operation again.

CTGKM1055W Deleting a key group deletes all the keys within a key group. Any data protected by these keys will be non-readable if the keys are deleted. Are you sure you would like to delete ${0}?

Explanation: Deleting a key group marks the key group and all keys in the group as destroyed in the database and deletes the material from the keystore.

System action: If you proceed, the key group and all keys in the key group are deleted.

Administrator response: Ensure that data protected by the keys in the key group is no longer needed. Then continue the delete operation.

CTGKM1056W Directory ${0} does not exist. Do you want to create it?

Explanation: The selected directory path that is specified to store the keystore does not currently exist.

System action: If you continue, the directory is created.

Administrator response: Confirm that you want to create the directory to store the keystore.

CTGKM1057W Deleting a key will render any data protected by the key non-readable. Are you sure you would like to delete the following key: ${0}?

Explanation: Deleting a key marks the key as destroyed in the database and deletes the material from the keystore. Data protected by the key is no longer readable.

System action: If you proceed, the key is deleted.

Administrator response: Ensure that data protected by the key is no longer needed. Then continue the delete operation.

CTGKM1058W Delete ${0}. Warning: Critical data will be deleted. Do you want to continue?

Explanation: You are about to delete a backup file. This message asks you to confirm the deletion.

System action: If you continue, the backup file is deleted.

Administrator response: Ensure that the data protected by this backup level is no longer needed, or is replicated in another level. Then continue the delete operation.

CTGKM1059E Path entered is not valid.

Explanation: The path that you entered is not a valid path.

System action: The path does not exist.

Administrator response: Specify a correct, existing path. Then try again.

CTGKM1060W Warning: If you delete ${0}, the level of this backup cannot be restored. Do you want to continue?

Explanation: Deleting a backup file means you cannot restore the level again.

System action: If you proceed, the backup file is deleted.

Administrator response: Ensure that data protected by this backup level is no longer needed, or is replicated in another level. Then continue the delete operation.

CTGKM1061W Creating backup to ${0}. Do you want to continue?

Explanation: This operation writes a backup file to the specified location.

System action: The operation writes a backup file to the location in this message.

Administrator response: Confirm that you want the backup file written to the specified location. Then continue.

CTGKM1062W The server will be stopped on all platforms except z/OS and the system will be restored from ${0}. IBM Security Key Lifecycle Manager key and configuration data will be restored to the level of the backup that you select. Any changes made after the selected backup will be lost, including the metadata. After restoring from this backup, you must manually restart the IBM Security Key Lifecycle Manager server. Do you want to continue?

Explanation: IBM Security Key Lifecycle Manager key and configuration data are restored to the level of the backup that you select. Any changes made after the selected backup are lost, including the metadata.

System action: Key and configuration data are restored to the level of the backup you select. Later changes are lost.

Administrator response: Confirm that you want to restore from this backup level. When the operation completes, manually restart the IBM Security Key Lifecycle Manager server.

CTGKM1063E A default key group needs to include at least one key.

Explanation: There are no keys in the default key group. Values in fields on this page might specify no keys, fail to add needed keys, or delete existing keys to zero.

System action: The operation fails.

Administrator response: Ensure that the key group has at least one valid key. Then try again to specify the key group as the default.

CTGKM1064E You cannot delete this certificate, which is currently used by ${0} tape drives as either a system default, or as a partner certificate. First, use the appropriate ${0} management panel to specify a different certificate as the system default or partner certificate. Then, delete the certificate.

Explanation: You cannot delete a certificate that is in use as the system default or partner certificate.

System action: The operation fails.

Administrator response: First, use the appropriate ${0} management panel to specify a different certificate as the system default or partner certificate. Then, delete the certificate.

CTGKM1065W The number of keys returned exceeds the limit of ${0} records, which are displayed. You might need to specify a different filter for your search. Then try again.

Explanation: The search returned more keys than the limit allows.

System action: The search fails.

Administrator response: You might need to specify a different filter for your search. Then try again.

CTGKM1066W Critical data has been added. Create a backup to ensure that you can restore this data.

Explanation: Critical data is added such as certificates or keys.

System action: Data is added.

Administrator response: To ensure that you can restore critical data in case of loss, create a backup file.

CTGKM1067W Are you sure you would like to delete ${0}? ${0} will no longer become a default in the future.

Explanation: This operation deletes assignment of the certificate or key group as a future default rollover.

System action: The system deletes the assignment as a rollover for the certificate or key group. The action does not delete the certificate or key group.

Administrator response: Determine that there is no requirement to use the certificate or key group as a future default. Then continue.

CTGKM1068E You need to select at least one type of default.

Explanation: You attempted to specify a certificate for future use as a rollover without also specifying that the certificate will be the default or partner certificate, or both, on an effective date.

System action: The operation pauses until you specify that the selected certificate is a default or partner certificate, or both, on the effective date.

Administrator response: Specify use as either the default or partner certificate, or both, for the certificate. Then continue.

CTGKM1069E Select a certificate from the list.

Explanation: To continue, the operation requires that you select a certificate.

System action: The operation pauses until you specify a certificate.

Administrator response: Select a valid certificate. Then continue.

CTGKM1070E There is no certificate defined. Close this panel and add a certificate.

Explanation: A required certificate is not in place.

System action: The operation fails.

Administrator response: Add the necessary certificate. Then try again.

CTGKM1071E IBM Security Key Lifecycle Manager application does not appear to be in a running state. The IBM Security Key Lifecycle Manager server may be down at the moment or is still starting. Make sure that the IBM Security Key Lifecycle Manager server is up and try your request again.

Explanation: The IBM Security Key Lifecycle Manager server might be down at the moment or is still starting.

System action: The operation fails.

Administrator response: Ensure that the IBM Security Key Lifecycle Manager server is running. Then try your request again.

CTGKM1075E Insufficient permission to see the information provided on this page.

Explanation: Your role does not have the necessary permission to view the data.

System action: The page does not display the information.

Administrator response: Obtain a user ID with the required permission. Then try again.

CTGKM1076W The current SSL/KMIP server certificate has been updated. Restart the server to put this change into effect. Then create a backup to ensure that you can restore this data.

Explanation: An update occurred to the current SSL/KMIP server certificate.

System action: The certificate update completes.

Administrator response: Restart the server to put this change into effect. Then create a backup to ensure that you can restore this data.

CTGKM1077W The current SSL server certificate has been updated. In order for this change to go into effect you must restart the server.

Explanation: An update occurred to the current SSL server certificate.

System action: The certificate update completes.

Administrator response: Restart the server to put this change into effect. Then create a backup to ensure that you can restore this data.

CTGKM1078W Are you sure you would like to delete certificate ${0}?

Explanation: Deleting a certificate marks the certificate as destroyed in the database and deletes the material from the keystore. Delete certificates only when the data protected by those certificates is no longer needed. Deleting certificates is similar to erasing the data. After certificates are deleted, data protected by those certificates is not retrievable.

System action: If you continue, the certificate is deleted.

Administrator response: Ensure that the certificate and the data that it protects are not needed. Then continue.

CTGKM1080E Insufficient permission to perform this action.

Explanation: Your role lacks one or more permissions required to take the action.

System action: The action fails.

Administrator response: Obtain a user ID with the required permissions. Then try again.

CTGKM1081W Are you sure that you want to delete device group ${0}?

Explanation: You might delete an empty device group such as myLTO. You cannot delete a device group if any devices, keys or certificates are in that group. You also cannot delete a device family that IBM Security Key Lifecycle Manager provides.

System action: If you continue, the device group is deleted.

Administrator response: Determine whether the empty device group is a valid candidate to delete. Then continue.

CTGKM1082E Select a certificate from the list.

Explanation: The operation requires that you specify a certificate.

System action: The operation waits for your selection.

Administrator response: Select a certificate. Then continue.

CTGKM1087W Are you sure you would like to reject device ${0}?

Explanation: The device is in a pending device list. You must accept or reject the device request to be served keys. You can only reject devices for device groups that you have permissions to create. By repeating a request, the device might appear again in the pending list.

System action: Rejection removes the device from the pending list.

Administrator response: Ensure that you do not want keys served to the device. Then reject the request.

CTGKM1088W Creating more than 1000 keys may take a few minutes. Do you want to continue?

Explanation: Creating a large number of keys requires a significant time interval.

System action: If you continue, IBM Security Key Lifecycle Manager creates the keys.

Administrator response: Ensure that you have time for this operation to complete. Then continue.

CTGKM1089E Insufficient permission to accept this device.

Explanation: You can only accept devices for device groups that you have permissions to create.

System action: The operation fails. Acceptance enables serving keys to the device and removes the device from the pending list.

Administrator response: Obtain a user ID that enables you to accept devices from the pending list. Then try again.

CTGKM1091E Unknown device family id: ${0}

Explanation: The device family for the operation is unknown.

System action: The operation fails.

Administrator response: Collect any information that might be in the audit log. You might need to contact Tivoli® Software Support.

CTGKM1092W Keystore file ${0} already exists. Would you like to use this existing keystore?

Explanation: The keystore name that you specified matches the name of an existing keystore.

System action: If you continue, the operation uses the existing keystore.

Administrator response: Determine whether the existing keystore is appropriate to use. Then continue.

CTGKM1093W No machines were found for the ${0} device family.

Explanation: Adding machines requires that you first specify whether device requests are automatically approved or held pending your approval.

System action: The operation cannot be done at this time.

Administrator response: To add machines, use the ${0} management panel to specify one of these choices:

- Automatically accept all new device requests for communication.
- Hold new device requests pending approval.

CTGKM1094W Do not use the setting of Automatically accept for the ${0} device family. This setting allows generation and serving of keys to ${0} storage servers before you can perform a backup.

Explanation: This setting is not appropriate for devices in this device family.

System action: The operation fails.

Administrator response: Select to manually add devices for communication, or to hold new device requests pending your approval. Then continue.

CTGKM1095W Changing the device group of a certificate from an ${0} causes the certificate to be unavailable to devices in other device groups that previously referenced the certificate. Are you sure you want to continue?

Explanation: This message occurs if you attempt to change (or update) an UNKNOWN certificate.

System action: If you continue, the assignment for this certificate is changed. This certificate will only be able to be used by devices in the selected device group or later modified to a different device group within the same device family.

Administrator response: If you are sure the certificate belongs to this device family, select **OK** to continue with the device group assignment.

CTGKM1100E Object (*object_type*) with identifier *object_id* cannot be found.

Explanation: The identifier value that you specified does not match an existing object.

System action: The operation fails.

Administrator response: Specify an identifier which corresponds to an existing object.

CTGKM1101E Incorrect or missing parameter was passed to perform a data store operation. Parameter identifier: *parameter_id*

Explanation: Incorrect input was specified for a data store operation.

System action: The operation fails.

Administrator response: Make sure that correct parameters are used as input.

CTGKM1107E Error: symmetricKeySet *VALUE_1* for parameter *VALUE_0* .

Explanation: The passed symmetricKeySet is either not a valid key or key group.

System action: The operation fails.

Administrator response: Specify a valid value for the parameter. Then, try the operation again.

CTGKM1108E Key Group for device group is not valid: *VALUE_1* for parameter *VALUE_0* .

Explanation: The passed symmetricKeySet key group is not the valid device group

System action: The operation fails.

Administrator response: Specify a valid value for the parameter. Then, try the operation again.

CTGKM1109E Key for device group is not valid *VALUE_1* for parameter *VALUE_0* .

Explanation: The passed symmetricKeySet key is not the valid device group.

System action: The operation fails.

Administrator response: Specify a valid value for the parameter. Then, try the operation again.

CTGKM1110E Certificate alias *VALUE_0* does not exist in this device group.

Explanation: The passed defaultAlias1/defaultAlias2 does not exist.

System action: The operation fails.

Administrator response: Specify a valid value for the parameter. Then, try the operation again.

CTGKM1111E Key Alias does not belong to device group *VALUE_1* for parameter *VALUE_0* .

Explanation: The passed defaultAlias1/defaultAlias2 does not match the given device group.

System action: The operation fails.

Administrator response: Specify a valid value for the parameter. Then, try the operation again.

CTGKM1115E Cannot change the device group of this device. It is associated with a key or key group.

Explanation: Cannot change the device group. The device is associated with a key or key group.

System action: Cannot change the device group. The device is associated with a key or key group.

Administrator response: Cannot change the device group. The device is associated with a key or key group.

CTGKM1116E Cannot change the device group of this device. The symmetric key alias of this device is being used as the default symmetric key alias of the device group, and cannot be moved.

Explanation: Cannot change the device group. The symmetric key alias of this device is being used as the default symmetric key alias of the device group, and cannot be moved.

System action: The operation fails.

Administrator response: Cannot change the device group. The symmetric key alias of this device is being used as the default symmetric key alias of the device group, and cannot be moved.

CTGKM1117E Cannot change the device group membership. The key is used by one or more devices.

Explanation: Cannot change the device group membership. The key is used by one or more devices.

System action: The operation fails.

Administrator response: Cannot change the device group membership. The key is used by one or more devices.

CTGKM1118E Cannot change the device group membership. The certificate is used by one or more devices.

Explanation: Cannot change the device group membership. The certificate is used by one or more devices.

System action: The operation fails.

Administrator response: Cannot change the device group membership. The certificate is used by one or more devices.

CTGKM1119E Cannot change the device group membership. The certificate is used by one or more devices.

Explanation: Cannot change the device group membership. The certificate is used by one or more devices.

System action: The operation fails.

Administrator response: Cannot change the device group membership. The certificate is used by one or more devices.

CTGKM1120E Cannot change the device group membership. The symmetric key is used by one or more devices.

Explanation: Cannot change the device group membership. The symmetric key is used by one or more devices.

System action: The operation fails.

Administrator response: Cannot change the device group membership. The symmetric key is used by one or more devices.

CTGKM1121E Cannot change the device group membership. The group is used by one or more devices.

Explanation: Cannot change the device group membership. The group is used by one or more devices.

System action: The operation fails.

Administrator response: Cannot change the device group membership. The group is used by one or more devices.

CTGKM1122E Cannot change the device group membership. The symmetric key is used by one or more devices.

Explanation: Cannot change the device group membership. The symmetric key is used by one or more devices.

System action: The operation fails.

Administrator response: Cannot change the device group membership. The symmetric key is used by one or more devices.

CTGKM1123E Cannot change the device group membership. The symmetric key is being used by another device group.

Explanation: Cannot change the device group membership. The symmetric key is being used by another device group.

System action: Cannot change the device group membership. The symmetric key is being used by another device group.

Administrator response: Cannot change the device group membership. The symmetric key is being used by another device group.

CTGKM1124E Cannot change the device group membership. The symmetric key is being used by another device.

Explanation: Cannot change the device group membership. The symmetric key is being used by another device.

System action: The operation fails.

Administrator response: Cannot change the device group membership. The symmetric key is being used by another device.

CTGKM1125E Cannot change the device group membership. The certificate is being used by another device group.

Explanation: Cannot change the device group membership. The certificate is being used by another device group.

System action: The operation fails.

Administrator response: Cannot change the device group membership. The certificate is being used by another device group.

CTGKM1126E Cannot change the device group membership. The certificate is being used by another device group.

Explanation: Cannot change the device group membership. The certificate is being used by another device group.

System action: The operation fails.

Administrator response: Cannot change the device group membership. The certificate is being used by another device group.

CTGKM1127E Cannot change the device group membership. The certificate is being used by another device.

Explanation: Cannot change the device group membership. The certificate is being used by another device.

System action: The operation fails.

Administrator response: Cannot change the device group membership. The certificate is being used by another device.

CTGKM1129E Target and source device groups family type does not match.

Explanation:

System action:

Administrator response:

CTGKM1130E Cannot delete a device group when keys, certificates, groups, or devices are attached to that device group.

Explanation:

System action:

Administrator response:

CTGKM1131E Key Alias Device Group does not match the device.

Explanation:

System action:

Administrator response:

CTGKM1132E Symmetric Key Alias device group does not match device.

Explanation:

System action:

Administrator response:

CTGKM1133E Cannot delete family device groups.

Explanation:

System action:

Administrator response:

CTGKM1134E Device group already exists.

Explanation:

System action:

Administrator response:

CTGKM1135E Incorrect device group family *VALUE_0*

Explanation: The device family must exist.

System action: The operation fails.

Administrator response: Specify an existing device family.

CTGKM1136E Group device group does not match the secret key device group usage.

Explanation: The group device group must match the secret key device group.

System action: The operation fails.

Administrator response: Group device group does not match the secret key device group usage.

CTGKM1137E Cannot add an empty key group as the default symmetricKeySet.

Explanation: The key group must have keys.

System action: The operation fails.

Administrator response: Cannot add an empty key group as the default symmetricKeySet.

CTGKM1138E No attributes were specified for the device group attribute update operation.

Explanation: No attribute-value pairs were specified to update information for a device group attribute.

System action: The operation fails.

Administrator response: Collect available audit log information and contact IBM Support.

CTGKM1139E Incorrect value for device group name *VALUE_0* \nThe device group name must follow the requirement: \n1. Can only contain alphanumeric characters and underscores. \n2. Cannot consist of a single underscore. \n3. First character cannot be a digit. \n4. Maximum length is 16.\n

Explanation: The device group name must follow the requirement: 1. Can only contain alphanumeric characters and underscore. 2. First character cannot be a digit. 3. Maximum length is 16.

System action: The operation fails.

Administrator response: The device group name must follow the requirement: 1. Name can only contain alphanumeric character and underscore 2. First character cannot be a digit. 3. Maximum length should be 16.

CTGKM1140E Device family *VALUE_0* cannot be used to create a device group.

Explanation: The specified device family cannot be used for this operation.

System action: Device group not created.

Administrator response: Specify a different device family. Then, try again.

CTGKM1141E aliasOne and aliasTwo are not valid attributes for the LTO device group and DS5000 device group.

Explanation: Key aliasOne and aliasTwo are not used by LTO and DS5000 device groups.

System action: Device not created or updated.

Administrator response: Remove the attributes aliasOne and aliasTwo. Then, try the operation again.

CTGKM1142E symAlias is not a valid attribute for a DS8000 device group and 3592 device group.

Explanation: symAlias is not used by DS8000 and 3592 device groups.

System action: Device not created or updated.

Administrator response: Remove the attribute symAlias. Then, try the operation again.

CTGKM1143E Cannot delete enableKMIPDelete attribute. enableKMIPDelete attribute must have a value of true or false.

Explanation: enableKMIPDelete attribute must have a value of true or false.

System action: The operation fails.

Administrator response: Do not run this command to delete the enableKMIPDelete attribute value.

CTGKM1144E Value for enableKMIPDelete attribute is not valid. A valid value is true or false.

Explanation: Value for enableKMIPDelete attribute is not valid. A valid value is true or false.

System action: The operation fails.

Administrator response: Specify true or false for the enableKMIPDelete attribute and try the operation again.

CTGKM1145E Expired or inactive Key Alias cannot be set as default.

Explanation: The defaultAlias1 or defaultAlias2 is expired or inactive

System action: The operation fails.

Administrator response: Specify a valid value for the parameter. Then, try the operation again.

CTGKM1146E The device group *VALUE_0* does not support secret keys.

Explanation: The device group does not support secret keys.

System action: The operation fails.

Administrator response: Specify a device group that supports secret keys. Then, try the operation again.

CTGKM1147E The device group *VALUE_0* does not support certificates.

Explanation: The device group does not support certificates.

System action: The operation fails.

Administrator response: Specify a device group that supports secret keys. Then, try the operation again.

CTGKM1148E Value for enableMachineAffinity attribute is not valid. A valid value is true or false.

Explanation: Value for enableMachineAffinity attribute is not valid. A valid value is true or false.

System action: The operation fails.

Administrator response: Specify true or false for the enableMachineAffinity attribute and try the operation again.

CTGKM1149E Value for device.AutoPendingAutoDiscovery attribute is not valid. Valid values are 0, 1 and 2.

Explanation: Value for device.AutoPendingAutoDiscovery attribute is not valid. Valid values are 0, 1 and 2.

System action: The operation fails.

Administrator response: Specify valid values (0,1,2) for device.AutoPendingAutoDiscovery attribute and try the operation again.

CTGKM1150E Cannot delete machineAffinity attribute. machineAffinity must have a value of true or false.

Explanation: machineAffinity attribute must have a value of true or false.

System action: The operation fails.

Administrator response: Do not run this command to delete the machineAffinity attribute value.

CTGKM1151E Cannot delete device.AutoPendingAutoDiscovery attribute. device.AutoPendingAutoDiscovery must have a value of 0, 1 or 2.

Explanation: device.AutoPendingAutoDiscovery attribute must have a value of 0, 1 or 2.

System action: The operation fails.

Administrator response: Do not run this command to delete the device.AutoPendingAutoDiscovery attribute value.

CTGKM1152E SSL or IKEv2-SCSI certificate is not allowed to move to other device groups.

Explanation: SSL or IKEv2-SCSI certificate is not allowed to move to other device groups.

System action: The operation fails.

Administrator response:

CTGKM1153E Cannot add a DS5000 family device with symAlias specified.

Explanation: You attempted to create a DS5000 family device and associate it with a key group.

System action: The device add operation fails.

Administrator response: Do not specify symAlias. Then, try the operation again.

CTGKM1154E Cannot associate a device with an empty key group.

Explanation: Cannot associate a device with an empty key group.

System action: The device update operation fails.

Administrator response: Specify a different symAlias. Then, try the operation again.

CTGKM1156E Conflicted key *VALUE_0* cannot be moved to a new device group.

Explanation: Conflicted key cannot be moved to a new device group.

System action: The key update operation fails.

Administrator response: Specify a different key alias. Then, try the operation again.

CTGKM1157E Unknown key can be moved to either *VALUE_0* or *VALUE_1* or *VALUE_2* device group only.

Explanation: Unknown key cannot be moved to a device group other than DS8000, 3592, SSLSERVER.

System action: The key update operation fails.

Administrator response: Specify a different device group. Then, try the operation again.

CTGKM1158E Conflicted certificate *VALUE_0* cannot be moved to a new device group.

Explanation: Conflicted certificate cannot be moved to a new device group.

System action: The certificate update operation fails.

Administrator response: Specify a different certificate alias. Then, try the operation again.

CTGKM1159E Unknown certificate can be moved to either *VALUE_0* or *VALUE_1* or *VALUE_2* device group only.

Explanation: Unknown certificate cannot be moved to a device group other than DS8000, 3592, SSLSERVER.

System action: The certificate update operation fails.

Administrator response: Specify a different device group. Then, try the operation again.

CTGKM1160E Unknown device can be moved to either *VALUE_0* or *VALUE_1* device group only.

Explanation: Unknown device cannot be moved to a device group other than LTO, 3592.

System action: The device update operation fails.

Administrator response: Specify a different device group. Then, try the operation again.

CTGKM1161E Conflicted certificate cannot be specified for a default rollover. This certificate is not valid: *alias*

Explanation: Conflicated certificate cannot be specified for a default rollover.

System action: The add rollover operation fails.

Administrator response: Specify a different certificate alias. Then, try the operation again.

CTGKM1162E The device group *VALUE_0* does not support keys.

Explanation: The device group does not support keys.

System action: The operation fails.

Administrator response: Specify a device group that support keys. Then, try the operation again.

CTGKM1163E The key with alias *VALUE_0* cannot be moved. Keys belonging to a key group cannot be moved between device groups.

Explanation: Keys are not allowed to be moved between device groups if that key is member of any key group.

System action: The key update operation fails.

Administrator response: Specify a key which does not belong to a key group, then try again.

CTGKM1200E Error getting device groups for device family *VALUE_0* .

Explanation: Exception occurred during the process.

System action: The operation fails.

Administrator response: See the log for more details.

CTGKM1201E Error getting device families for device group *VALUE_0* .

Explanation: Exception occurred during the process.

System action: See the log for more details.

Administrator response: See the log for more details.

CTGKM1202E Error getting a list of device groups.

Explanation: Exception occurred during the process.

System action: See the log for more details.

Administrator response: See the log for more details.

CTGKM1203E Error creating device group *VALUE_0* .

Explanation: Exception occurred during the process.

System action: See the log for more details.

Administrator response: See the log for more details.

CTGKM1204E Error deleting device group *VALUE_0* .

Explanation: Exception occurred during the process.

System action: See the log for more details.

Administrator response: See the log for more details.

CTGKM1205E Error setting machine affinity.

Explanation: Exception occurred during the process.

System action: See the log for more details.

Administrator response: See the log for more details.

CTGKM1206E Error setting auto pending.

Explanation: Exception occurred during the process.

System action: See the log for more details.

Administrator response: See the log for more details.

CTGKM1207E Error getting devices referencing key group.

Explanation: Exception occurred during the process.

System action: See the log for more details.

Administrator response: See the log for more details.

CTGKM1208E Error getting devices referencing certificate.

Explanation: Exception occurred during the process.

System action: See the log for more details.

Administrator response: See the log for more details.

CTGKM1209E Concurrent update error. Another user might have changed the data.

Explanation: Exception occurred while performing a concurrent update.

System action: See the log for more details.

Administrator response: Refresh and try again.

CTGKM1210E An error occurred while accepting the pending device.

Explanation: The device may have already been accepted, there may be an error in the additional fields specified while accepting this device, or the IBM Security Key Lifecycle Manager database might not be available. Additional information should accompany this message.

System action: The device may not have been accepted, or the device was accepted but additional fields specified for the device may not have been set.

Administrator response: The additional information accompanying this message might guide your response. You might need to confirm that the database is available.

CTGKM1211E An error occurred while rejecting the pending device.

Explanation: The device may have already been rejected by another user or the IBM Security Key Lifecycle Manager database might not be available. Additional information should accompany this message.

System action: The device may not have been rejected.

Administrator response: The additional information accompanying this message might guide your response. You might need to confirm that the database is available.

CTGKM1212E An error occurred while setting the default key group.

Explanation: IBM Security Key Lifecycle Manager database might not be available. Additional information should accompany this message.

System action: Cannot update the default key group.

Administrator response: The additional information accompanying this message might guide your response.

CTGKM1213E An error occurred while setting the default certificate.

Explanation: IBM Security Key Lifecycle Manager database might not be available. Additional information should accompany this message.

System action: Cannot update the default certificate.

Administrator response: The additional information accompanying this message might guide your response.

CTGKM1214E An error occurred while setting the partner certificate.

Explanation: IBM Security Key Lifecycle Manager database might not be available. Additional information should accompany this message.

System action: Cannot update the partner certificate.

Administrator response: The additional information accompanying this message might guide your response.

CTGKM1215W Not all keys were made. Some aliases of the keys for the specified key group collided with another key group generated at the same time. You might want to use Modify Key Group panel to add additional keys.

Explanation: There were key alias conflicts and the total number of keys requested could not be created.

System action: No action required.

Administrator response: Use Modify Key Group panel to add additional keys.

CTGKM1301E The key group *VALUE_0* cannot be set as a system or device default, because all its keys are compromised.

Explanation: Key groups containing only compromised keys cannot be set as a system or device default.

System action: The operation fails.

Administrator response: Add non-compromised keys to the key group, or specify a different key group.

CTGKM1302E The private key algorithm is *VALUE_0*, which is not supported for usage *VALUE_1*. The supported algorithms are: *VALUE_2*

Explanation: The private key uses an encryption algorithm which is not supported for the specified device group.

System action: The operation fails.

Administrator response: Change the usage, or specify another private key with an appropriate encryption algorithm. Then, try the operation again.

CTGKM1303E Unable to access keystore file *VALUE_0*.

Explanation: Keystore file is missing or not readable.

System action: The operation fails.

Administrator response: Ensure that the specified file exists and has the correct permissions.

CTGKM1307E Keys could not be released because no backup has been made.

Explanation: A backup is required before keys can be released.

System action: The operation fails.

Administrator response: Make a backup, then try the operation again.

CTGKM1308E The configuration property *VALUE_0* cannot be manually updated or deleted.

Explanation: The configuration property cannot be manually updated or deleted.

System action: The operation fails.

Administrator response: No action required.

CTGKM1400E Machine ID cannot be null.

Explanation: Machine ID is a required attribute.

System action: The operation fails.

Administrator response: Specify a valid machine ID.

CTGKM1401E Either machine ID or machine text is required.

Explanation: Either machine ID or machine text is required.

System action: The operation fails.

Administrator response: Specify either the machine ID or machine text. Then, try the operation again.

CTGKM1402E Machine UUID cannot be null.

Explanation: Machine UUID cannot be a null attribute.

System action: The operation fails.

Administrator response: Specify a value for the machine UUID.

CTGKM1403E Device group does not exist.

Explanation: The specified device group is not a valid or known device group.

System action: The operation fails.

Administrator response: Specify another valid device group.

CTGKM1404E Device group is not a DS5000 device group.

Explanation: The specified device group is not a DS5000 device group or a member of the DS5000 device family.

System action: The operation fails.

Administrator response: Specify a valid DS5000 device group or DS5000 device family.

CTGKM1405E Device UUID cannot be null.

Explanation: The specified device UUID cannot be empty or null.

System action: The operation fails.

Administrator response: Specify a valid device UUID.

CTGKM1406E Device does not exist.

Explanation: The specified device does not exist or is not a valid device in IBM Security Key Lifecycle Manager.

System action: The operation fails.

Administrator response: Specify a valid and known device group.

CTGKM1407E No machine IDs exist.

Explanation: No machine IDs exist in IBM Security Key Lifecycle Manager.

System action: The operation fails.

Administrator response: No machine IDs exist.

CTGKM1408E Machine ID/Text not found.

Explanation: This Machine ID/Text does not exist or is not valid.

System action: The operation fails.

Administrator response: Specify a valid machine identifier.

CTGKM1409E Machine affinity is ON. The device is associated with at least one machine and cannot be deleted or rejected.

Explanation: Machine affinity is ON. The device is associated with at least one machine and cannot be deleted or rejected.

System action: The operation fails.

Administrator response: Delete the machine device association before trying to delete or reject the device.

CTGKM1410E Either Machine UUID, Machine Text, or Machine ID is required.

Explanation: A value is required for either machine UUID, machine text, or machine ID.

System action: The operation fails.

Administrator response: Specify the machine UUID, machine text, or machine ID, and try again.

CTGKM1411E Machine Identifier field must be between 1 (minimum) and 48 (maximum) characters in length.

Explanation: Machine Identifier field should be between 1 and 48 characters in length.

System action: The operation fails.

Administrator response: Specify a machine identifier that is between 1 (minimum) and 48 (maximum) characters in length.

CTGKM1416E The specified machine must exist for the operation to succeed.

Explanation: Machine device association cannot be added. The machine does not exist.

System action: The operation fails.

Administrator response: Specify a valid machine identifier for this association.

CTGKM1417E Machine does not exist and cannot be updated.

Explanation: Machine lookup failed. It appears the machine does not exist.

System action: The operation fails.

Administrator response: Specify a valid machine identifier to update.

CTGKM1418E Machine text is not unique.

Explanation: Machine text is not unique. There is already a machine ID with that machine text.

System action: The operation fails.

Administrator response: Specify a unique string for the machine text.

CTGKM1419E Machine has existing machine affinities.

Explanation: The machine has existing machine affinities.

System action: The operation fails.

Administrator response: Delete all machine affinities before deleting the machine ID.

CTGKM1420E Device text is not unique.

Explanation: There is already a device with that device text.

System action: The operation fails.

Administrator response: Specify a different unique device text.

CTGKM1422E serialNumber cannot be updated with the device group.

Explanation: A serial number update is not supported for the device group.

System action: The operation fails.

Administrator response: Do not use a serial number as an attribute for this action.

CTGKM1423E *VALUE_0* **is not allowed for a non-DS5000 device group.**

Explanation: This attribute is not allowed for a non-DS5000 device group.

System action: The operation fails.

Administrator response: Do not use this attribute for this action.

CTGKM1425E The machine device association or the device is not pending.

Explanation: The machine device association or the device has been accepted or is not in pending status.

System action: No action taken.

Administrator response: You cannot accept a nonpending machine/device.

CTGKM1426E Machine ID is not unique.

Explanation: The machine ID given is not unique in IBM Security Key Lifecycle Manager.

System action: The operation fails.

Administrator response: Specify a valid unique machine identifier.

CTGKM1427E Machine affinity already exists.

Explanation: Machine affinity for this device and machine already exists in IBM Security Key Lifecycle Manager.

System action: The operation fails.

Administrator response: Specify a different unique machine identifier and device.

CTGKM1429E The DS5000 number of keys cannot exceed 12.

Explanation: The DS5000 number of keys cannot exceed 12.

System action: The operation fails.

Administrator response: Specify a value between 0 and 12 for the number of keys.

CTGKM1430E This device has pending machine affinities. Reject the pending machine affinities first.

Explanation: This pending device has pending machine affinities. Reject the pending machine affinities first.

System action: The operation fails.

Administrator response: Reject the pending machine affinities first.

CTGKM1431E Value for DS5000 number of keys is not valid. The value must be a positive integer.

Explanation: The value for the DS5000 number of keys can only be a positive integer.

System action: The operation fails.

Administrator response: Specify a positive integer for the number of keys.

CTGKM1432E symmetricKeySet is not a valid attribute for a DS5000 device group.

Explanation: symmetricKeySet is not used by DS5000.

System action: Device group not created or updated.

Administrator response: Remove the symmetricKeySet attribute and try the operation again.

CTGKM1433E Values for machineText *VALUE_1* **and machineID** *VALUE_0* **do not match.**

Explanation: MachineText and MachineID do not match.

System action: Machine device will not be listed.

Administrator response: Specify a different value for machineText or machineID and try the operation again.

CTGKM1434E Machine Text must be between 1 and 96 characters in length.

Explanation: The value of the machineText parameter must be between 1 and 96 characters in length.

System action: The operation fails.

Administrator response: Specify a value for the machineText parameter between 1 and 96 characters in length.

CTGKM1435E Machine ID/Text does not match the Machine UUID.

Explanation: Machine ID/Text does not match the machine with the specified UUID.

System action: The operation fails.

Administrator response: Specify a different value for Machine ID/Text.

CTGKM1436E No machine affinity between device *VALUE_0* **and machine** *VALUE_1*.

Explanation: Machine affinity is only supported for DS5000 devices.

System action: The operation fails.

Administrator response: Specify a DS5000 device.

CTGKM1500E NULL attribute array.

Explanation: No attributes were specified for a new template.

System action: The template is not created.

Administrator response: Specify attributes for the template, then try the operation again.

CTGKM1501E Found inappropriate attribute for template: *VALUE_0*

Explanation: The specified attribute is not supported by the template.

System action: The template is not created.

Administrator response: Remove or change the unsupported attribute, then try the operation again.

CTGKM1502E NULL template names array.

Explanation: No template names were specified to be merged.

System action: The template merge fails.

Administrator response: Specify template names to be merged, then try the operation again.

CTGKM1503E Template with name *VALUE_0* not found.

Explanation: The specified template was not found in the database.

System action: The message processing fails because the template attributes cannot be read.

Administrator response: Specify the correct template name and try again.

CTGKM1504E The authentication information in the request was not able to be validated, or there was no authentication information in the request when there SHOULD have been.

Explanation: The authentication information is either missing or not valid.

System action: The authentication failed because the information provided is not valid or missing.

Administrator response: Specify the correct authentication information and try again.

CTGKM1505E The client does not have permission to perform the requested operation.

Explanation: The client is not authorized to perform the requested operation.

System action: The requested operation failed because the client does not have the permission.

Administrator response: Make sure that the requested operation is authorized and retry the action.

CTGKM1506E The operation failed due to a cryptographic error.

Explanation: The requested operation failed due to a cryptographic error.

System action: The requested operation failed because the key cannot be read due to a cryptographic error.

Administrator response: Please look at the exception message and take appropriate action.

CTGKM1507E An OPTIONAL feature specified in the request is not supported.

Explanation: The requested OPTIONAL feature is not supported.

System action: The request failed because the feature is not supported.

Administrator response: Remove or change the feature, then try the operation again.

CTGKM1508E The client requested an operation that was not able to be performed with the specified parameters.

Explanation: The specified parameters are not valid for the requested operation.

System action: The requested operation failed because the specified parameters are not valid.

Administrator response: Specify the correct parameters and try again.

CTGKM1509E Some data item in the request has an incorrect value.

Explanation: Some of the parameter value in the request is not valid.

System action: The requested operation failed because one or more of the attributes has an incorrect value.

Administrator response: Make sure to pass in the correct attribute values and retry the action.

CTGKM1510E The request message was not understood by the server.

Explanation: The message in the request was not understood by the server.

System action: The processing of the message failed because it was not understood by the server.

Administrator response: Specify the correct message in the request and try again.

CTGKM1511E A requested object was not found or did not exist.

Explanation: The requested object was not found.

System action: The requested operation failed because the object did not exist.

Administrator response: Make sure to pass the identifier for the existing object and retry the action.

CTGKM1512E The operation requires additional OPTIONAL information in the request, which is not present.

Explanation: The request is missing the OPTIONAL information required for the operation.

System action: The requested operation failed because it is missing the OPTIONAL information required.

Administrator response: Make sure to pass in the required OPTIONAL information and retry the action.

CTGKM1513E The object must be recovered from the archive before performing the operation.

Explanation: The requested object is archived.

System action: The requested operation failed because the object is archived.

Administrator response: Make sure that the object is recovered and retry the action.

CTGKM1514E The operation was asynchronous, and the operation was canceled by the Cancel operation before it completed successfully.

Explanation: The asynchronous operation was canceled before it completed successfully.

System action: The requested operation failed because it was canceled before it completed successfully.

Administrator response: No action required.

CTGKM1515E The operation requested by the request message is not supported by the server.

Explanation: The requested operation is not supported by the server.

System action: The requested operation failed because it is not supported by the server.

Administrator response: Specify a supported operation then try the operation again.

CTGKM1516E The response to a request would exceed the maximum response size in the request.

Explanation: The response size exceeds the maximum response size specified in the request.

System action: The requested operation failed because the response size exceeds the maximum response size specified in the request.

Administrator response: Increase the maximum response size in the request, then try the operation again.

CTGKM1517E Value out of range.

Explanation: The specified value is out of range.

System action: The requested operation failed because the value specified is out of the enumerated list.

Administrator response: Specify the correct value and retry the action.

CTGKM1520E The server was not able to perform the requested operation.

Explanation: An unexpected error occurred on the KMIP server.

System action: This error is returned to the KMIP client.

Administrator response: Please look at the exception message and take appropriate action.

CTGKM1521E The operation failed due to an inappropriate index.

Explanation: The caller passed an incorrect index on a multivalued attribute.

System action: This error is returned to the KMIP client.

Administrator response: Specify another index, and try the operation again.

CTGKM1522E Attribute *VALUE_0* **is not supported.**

Explanation: Attribute name is unknown or unsupported.

System action: The operation fails.

Administrator response: Specify a supported attribute name, and try the operation again.

CTGKM1523E Index must be specified for update or delete operation on a multivalued attribute.

Explanation: Index was not specified for an update or delete operation on a multivalued attribute.

System action: The operation fails.

Administrator response: Specify an index, and try the operation again.

CTGKM1524E The operation *VALUE_0* **is not supported.**

Explanation: The operation is not supported.

System action: The operation fails.

Administrator response: Specify a supported operation name, then try again.

CTGKM1525E The field *VALUE_0* is not supported for the attribute *VALUE_1*.

Explanation: The field provided is not supported for the attribute.

System action: The operation fails.

Administrator response: Specify a field that is supported for the attribute, then try the operation again.

CTGKM1526E Date must be in the format *VALUE_0*, and represent a valid date.

Explanation: The date supplied is not valid.

System action: The operation fails.

Administrator response: Specify a valid date in the correct format, then try the operation again.

CTGKM1527E Attribute values must be specified for add or update operation.

Explanation: The add and update attributes operations require that attribute values be specified.

System action: The operation fails.

Administrator response: Specify the attribute values, then try the opreation again.

CTGKM1528E *VALUE_0* fields must be specified for attribute *VALUE_1*.

Explanation: The listed fields are required when adding or updating this attribute.

System action: The operation fails.

Administrator response: Specify values for all the listed fields, then try the operation again.

CTGKM1529E Name for a custom attribute must begin with either *VALUE_0* or *VALUE_1*.

Explanation: The name for a custom KMIP attribute must follow the stated requirements.

System action: The operation fails.

Administrator response: Specify a custom attribute name that meets the requirements, then try the operation again.

CTGKM1530E The value *VALUE_0* is not supported for the field *VALUE_1*.

Explanation: The value is not supported for the specified field.

System action: The operation fails.

Administrator response: Specify a supported value, then try the operation again.

CTGKM1531E The value exceeds the limit for a multi-valued attribute.

Explanation: The value exceeds the limit for a multi-valued attribute set by the IBM Security Key Lifecycle Manager server.

System action: The operation fails.

Administrator response: Set the property mv.attribute.max.values in SKLMConfig.properties file to a higher value and try again.

CTGKM1532E Value contains reserved wildcard characters that are not allowed.

Explanation: The specified value contains reserved wildcard characters such as (* and %).

System action: The requested operation failed because the value specified is not valid.

Administrator response: Specify the correct value and retry the action.

CTGKM1533E The request cannot be processed.

Explanation: The request from this device cannot be processed.

System action: The requested operation fails.

Administrator response: Check the setting for accepting devices for this device group through the graphical user interface. Take appropriate action either to accept this device from the pending list or set the flag to automatically accept all new devices. Then try the request again.

CTGKM1534E The request cannot be processed.

Explanation: An internal error occurred and the request from this device cannot be processed.

System action: The requested operation fails.

Administrator response: Check the setting for accepting devices for this device group through the graphical user interface. Then retry the action.

CTGKM1535E *VALUE_0* is too large.

Explanation: The object or attribute is oversized and cannot be processed.

System action: The operation fails.

Administrator response: The key bytes length exceeds the maximum length 8K supported by the keystore. Try again with a reduced size.

CTGKM1536E Key must be specified.

Explanation: The key to create or register is not specified in the request.

System action: The requested operation fails.

Administrator response: Specify the key, then try the operation again.

CTGKM1537E Key group *VALUE_0* does not exist.

Explanation: The specified key group in the request does not exist.

System action: The requested operation fails.

Administrator response: Specify an existing key group, then try the operation again.

CTGKM1538E Usage mask must be specified.

Explanation: The usage mask must be specified for a symmetric key in the request.

System action: The requested operation fails.

Administrator response: Specify the usage mask, then try the operation again.

CTGKM1539E Algorithm *VALUE_0* not supported.

Explanation: Algorithm not supported for the requested operation.

System action: The requested operation fails.

Administrator response: Specify a supported algortihm, then try the operation again.

CTGKM1540E Key size *VALUE_0* not supported for algorithm *VALUE_1*.

Explanation: Key size is not supported for the algorithm specified in the request.

System action: The requested operation fails.

Administrator response: Ensure that the key size specified is supported by the algorithm. Try the operation again.

CTGKM1541E Unexpected key type, only register of SecretKey supported.

Explanation: To register a key, it must be of type javax.crypto.SecretKey in the request.

System action: The requested operation fails.

Administrator response: Specify a key of type SecretKey, then try the operation again.

CTGKM1542E JCE problem with DESede or AES while generating a secret key.

Explanation: JCE unable to generate a secret key with algorithm DESede or AES.

System action: The requested operation fails.

Administrator response: Ensure that the JCE provider supports the requested algorithm and try again.

CTGKM1543E Algorithm *VALUE_0* not supported by the provider for hashing.

Explanation: The algorithm is not supported by the provider.

System action: The requested operation fails.

Administrator response: Specify an algorithm that is supported by the provider, then try the operation again.

CTGKM1544E The following attribute(s) are not allowed for the *VALUE_0* operation: *VALUE_1*

Explanation: Some of the attributes are not allowed for the requested operation.

System action: The requested operation fails.

Administrator response: Remove the attributes that are not allowed, and try the operation again.

CTGKM1545E y-KeyGroupGetNext must supply a key group name.

Explanation: The y-KeyGroupGetNext custom server attribute must supply a key group name.

System action: The requested operation fails.

Administrator response: Specify a key group name, then try the operation again.

CTGKM1546E An object with the nametype of *VALUE_0* and the namevalue of *VALUE_1* already exists.

Explanation: The caller is trying to reuse an existing name for an object.

System action: The requested operation fails.

Administrator response: Specify a different nametype/namevalue. Then try the operation again.

CTGKM1547E This *VALUE_0* requires an instance of *VALUE_1*.

Explanation: The caller is trying to set an unsupported value.

System action: The requested operation fails.

Administrator response: Specify a supported value, then try the operation again.

CTGKM1548E Unsupported object type: *VALUE_0*

Explanation: The caller is requesting an operation on an unsupported object type.

System action: The requested operation fails.

Administrator response: Specify a supported object type, then try the operation again.

CTGKM1549E StorageStatusMask: *VALUE_0* **is not valid.**

Explanation: The caller is requesting an unsupported storage status mask.

System action: The requested operation fails.

Administrator response: Specify a supported storage status mask and try the operation again.

CTGKM1550E No search attributes were specified on the LOCATE request.

Explanation: The caller omitted required parameters.

System action: The requested operation fails.

Administrator response: Specify the search attributes in the LOCATE request, then try again.

CTGKM1551E The *VALUE_0* **operation is not valid for an object of type** *VALUE_1*.

Explanation: The caller requested an operation that is inappropriate for that object.

System action: The requested operation fails.

Administrator response: Specify an operation supported for the object, then try again.

CTGKM1552E The *VALUE_0* **attribute requires a non-null value.**

Explanation: The attribute cannot contain a null value.

System action: Cannot process the message.

Administrator response: Ensure that the attribute value is non-null, then try again.

CTGKM1553E The attribute name is not specified.

Explanation: The attribute name is not specified.

System action: The requested operation fails.

Administrator response: Specify the attribute name, then try the operation again.

CTGKM1554E The attribute *VALUE_0* **does not exist.**

Explanation: The specified attribute does not exist.

System action: The requested operation fails.

Administrator response: Specify an existing attribute, then try the operation again.

CTGKM1555E The index *VALUE_0* **is not valid for a single-valued attribute.**

Explanation: For a single-valued attribute, the index can only be 0.

System action: The requested operation fails.

Administrator response: Specify the correct index for a single-valued attribute, then try the operation again.

CTGKM1556E The value at index *VALUE_0* **does not exist.**

Explanation: The value at the specified index does not exist.

System action: The requested operation fails.

Administrator response: Ensure that a value exists at the specified index. Then, try the operation again.

CTGKM1557E Could not construct *VALUE_0* **from the input provided. Underlying field name or error message** *VALUE_1*

Explanation: The caller provided input that could not be processed.

System action: The requested operation fails.

Administrator response: Specify a correct value for this attribute, then try the operation again.

CTGKM1558E No key material is available for the object with identifier *VALUE_0*.

Explanation: No key material is available, possibly because none was ever sent to the server.

System action: The requested operation fails.

CTGKM1559E No key material is available for the object with identifier *VALUE_0*. **The object has been destroyed.**

Explanation: No key material exists on the server because the object has been destroyed.

System action: The requested operation fails.

CTGKM1560E Object with UUID *VALUE_0* and type *VALUE_1* does not exist.

Explanation: An object with the specified UUID and type does not exist.

System action: The requested operation fails.

Administrator response: Specify another UUID and type, then try the operation again.

CTGKM1561E Object with UUID *VALUE_0* could not be served for cryptographic use because it is not backed up.

Explanation: Object with the specified UUID cannot be served for cryptographic use because it is not backed up.

System action: The requested operation fails.

Administrator response: Back up IBM Security Key Lifecycle Manager, then try the operation again.

CTGKM1562E Cannot register a key in a key group if no key material is supplied by the caller.

Explanation: Keys that are in key groups must contain cryptographic material, but the caller is not providing any.

System action: The requested operation fails.

CTGKM1563E Object with UUID *VALUE_0* could not be served for cryptographic use because it is not released.

Explanation: Object with the specified UUID cannot be served for cryptographic use because it is not released. This message may also occur if the config property release.date is missing or wrongly formatted.

System action: The requested operation fails.

Administrator response: Run the tklmKeyRelease command to release keys, then try the operation again.

CTGKM1701E For DS5000 device group, device text must be less than 96 characters in length.

Explanation: For the DS5000 device group, the value of the deviceText parameter must be less than 96 characters in length.

System action: The operation fails.

Administrator response: For DS5000 devices, specify a value for deviceText that is less than 96 characters in length.

CTGKM1702E Unable to generate more than 12 keys at a time for DS5000 group.

Explanation: DS5000 keys can only generated 12 or less at a time.

System action: The operation fails.

Administrator response: Reduce the number of requested keys and try the operation again.

CTGKM1703E Incorrect device group ID : *VALUE_0*

Explanation: Device group ID is not found.

System action: The operation fails.

Administrator response: Change the device group and try the operation again.

CTGKM1704E Device description cannot exceed 255 characters in length.

Explanation: The device description exceeded 255 characters in length.

System action: The operation fails.

Administrator response: Change to a shorter description and try the operation again.

CTGKM1706E The certificate *VALUE_0* is scheduled for a future rollover, and cannot be moved or deleted.

Explanation: A certificate which is scheduled for a future rollover cannot be moved or deleted.

System action: The operation fails.

Administrator response: Remove any pending rollovers for this certificate, then try the operation again.

CTGKM1707E The key group *VALUE_0* is scheduled for a future rollover, and cannot be moved or deleted.

Explanation: A key group which is scheduled for a future rollover cannot be moved or deleted.

System action: The operation fails.

Administrator response: Remove any pending rollovers for this key group, then try the operation again.

CTGKM1901E Device group: *VALUE_0* is not a valid group for license count.

Explanation: This device group must be one of the valid device groups listed in the license list.

System action: Specify a correct group name.

Administrator response: Specify a correct group name.

CTGKM1936E dateAfter cannot be later than dateBefore.

Explanation: dateAfter is after dateBefore. This will give an empty time interval.

System action: The operation fails.

Administrator response: Specify different dateAfter and dateBefore values, and try the operation again.

CTGKM2100E The value for replication.role is not valid. Accepted values are CLONE or MASTER.

Explanation: The value for the replication.role must be CLONE or MASTER.

System action: The operation fails.

Administrator response: Correct setting of replication.role parameter to CLONE or MASTER.

CTGKM2102E No valid replication config file exists. It will be created.

Explanation: No valid replication config file exists. It will be created.

System action: The operation fails.

Administrator response: No action required.

CTGKM2102E No valid replication config file exists. It will be created.

Explanation: No valid replication config file exists. It will be created.

System action: The operation fails.

Administrator response: No action required.

CTGKM2103E The value for replication.MaxLogFileSize is not valid. Acceptable values are between 100 and 500000 bytes.

Explanation: The replication log file size can be between 100 and 500000 bytes.

System action: The operation fails.

Administrator response: Correct setting of replication.MaxLogFileSize.

CTGKM2104E The value for replication.MaxLogFileNum is not valid. Acceptable values are between 2 and 100.

Explanation: The value for replication.MaxLogFileNum must be between 2 and 100.

System action: The operation fails.

Administrator response: Correct setting of replication.MaxLogFileNum.

CTGKM2105E The value for replication.MaxBackupNum is not valid. Acceptable values are between 2 and 10.

Explanation: The value for replication.MaxBackupNum must be between 2 and 10.

System action: The operation fails.

Administrator response: Correct setting of replication.MaxBackupNum.

CTGKM2106E Value for replication.MasterListenPort is not valid. Acceptable values are integers between 1 and 65535.

Explanation: Valid values for replication.MasterListenPort are integers between 1 and 65535.

System action: The operation fails.

Administrator response: Correct setting of replication.MasterListenPort.

CTGKM2107E Value for replication.MasterListenPort is not a valid port number or is used as a port elsewhere in IBM Security Key Lifecycle Manager.

Explanation: replication.MasterListenPort is not a valid port number, or is the same as one of restore.ListenPort, TransportListener.tcp.port, KMIPListener.ssl.port or TransportListener.ssl.port.

System action: The operation fails.

Administrator response: Correct setting of replication.MasterListenPort to a valid, available port number.

CTGKM2108E Value for the backup.CheckFrequency is not valid. Acceptable values are integers equal to or greater than 1.

Explanation: Valid values for backup.CheckFrequency are integers equal to or greater than 1.

System action: The operation fails.

Administrator response: Correct setting of backup.CheckFrequency to be an integer equal to or greater than 1.

CTGKM2109E Command ignored. Backup time has already been set.

Explanation: Command ignored. Backup time has already been set as per the backup.DailyStartReplicationBackupTime parameter.

System action: Command ignored.

Administrator response: No action required.

CTGKM2110E Value of backup.DailyStartReplicationBackupTime is not valid. It should be in HH:MM format.

Explanation: Value of backup.DailyStartReplicationBackupTime should be a valid time in HH:MM format.

System action: The operation fails.

Administrator response: Correct backup.DailyStartReplicationBackupTime to be a valid time in HH:MM format.

CTGKM2111E Value for backup.SerializeRestores is not valid. Accepted values are either true or false.

Explanation: Value for configuration parameter backup.SerializeRestores must be either true or false.

System action: The operation fails.

Administrator response: Correct backup.SerializeRestores to be either true or false.

CTGKM2112E The value for backup.BackupDescriptionText must not exceed 100 characters.

Explanation: The value for backup.BackupDescriptionText must not exceed 100 characters.

System action: The operation fails.

Administrator response: Correct backup.BackupDescriptionText such that it does not exceed 100 characters.

CTGKM2113E The value for backup.ReleaseKeys is not valid. Accepted values are RESTORE, BACKUP or OFF.

Explanation: The value for backup.ReleaseKeys must be one of RESTORE, BACKUP or OFF.

System action: The operation fails.

Administrator response: Correct backup.ReleaseKeys to be one of RESTORE, BACKUP or OFF.

CTGKM2114E Command ignored. To update this property, make sure the value of the enableKeyRelease parameter in the IBM Security Key Lifecycle Manager configuration file has been set to true.

Explanation: In order to update the backup.ReleaseKeys, the value of the enableKeyRelease parameter in the IBM Security Key Lifecycle Manager configuration file should have been already set to true.

System action: The operation fails.

Administrator response: No action required.

CTGKM2115E restore.ListenPort parameter is not a valid port number. Accepted values are integers between 1 and 65535.

Explanation: The value of restore.ListenPort parameter should be an integer between 1 and 65535.

System action: The operation fails.

Administrator response: Correct restore.ListenPort to be a valid port number.

CTGKM2116E The value of restore.ListenPort must not be shared with any other IBM Security Key Lifecycle Manager port parameter settings.

Explanation: The value for the restore.ListenPort parameter cannot be the same as values for the replication.MasterListenPort, TransportListener.tcp.port, KMIPListener.ssl.port or TransportListener.ssl.port.

System action: The operation fails.

Administrator response: Correct restore.ListenPort to be a valid port number not used elsewhere in IBM Security Key Lifecycle Manager.

CTGKM2117E Value of restore.DailyStartReplicationRestoreTime is not valid. It should be in HH:MM format.

Explanation: The value for restore.DailyStartReplicationRestoreTime should be a valid time in HH:MM format.

System action: The operation fails.

Administrator response: Correct restore.DailyStartReplicationRestoreTime to be a valid time in HH:MM format.

CTGKM2118E The value for the restore.NumAttemptRetryFailedRestore parameter must be between 0 and 2.

Explanation: The value for the

restore.NumAttemptRetryFailedRestore parameter must be between 0 and 2.

System action: The operation fails.

Administrator response: Correct restore.NumAttemptRetryFailedRestore to be between 0 and 2.

CTGKM2119E Value for restore.RevertToPreviousBackupOnFailure must be either true or false.

Explanation: Value for configuration parameter restore.RevertToPreviousBackupOnFailure can only be either true or false.

System action: The operation fails.

Administrator response: Correct restore.RevertToPeviousBackupOnFailure to be either true or false.

CTGKM2120E Value for the backup.ClientPort(n) parameter must be an integer between 1 and 65535.

Explanation: Value for the backup.ClientPort(n) parameter must be an integer between 1 and 65535.

System action: The operation fails.

Administrator response: Correct backup.ClientPort(n) to be an integer between 1 and 65535.

CTGKM2121E backup.ClientPort must not be shared with any other IBM Security Key Lifecycle Manager port parameter settings.

Explanation: The value for the backup.ClientPort parameter cannot be the same as values for the replication.MasterListenPort, TransportListener.tcp.port, KMIPListener.ssl.port or TransportListener.ssl.port.

System action: The operation fails.

Administrator response: Correct restore.ListenPort to be a valid port number not used elsewhere in IBM Security Key Lifecycle Manager.

CTGKM2122E backup.EncryptionPassword must not be fewer than 6 characters or exceed 175 single-byte or 87 double-byte characters.

Explanation: backup.EncryptionPassword cannot be null, fewer than 6 characters or longer than 175 single-byte or 87 double-byte characters.

System action: The operation fails.

Administrator response: Correct backup.EncryptionPassword to a valid value.

CTGKM2123E The backup.ObfuscatedEncryptionPassword parameter cannot be updated.

Explanation: The backup.ObfuscatedEncryptionPassword parameter cannot be updated.

System action: The operation fails.

Administrator response: No action required.

CTGKM2124E The StopReplication has timed out!.

Explanation: The StopReplication has timed out!.

System action: The operation fails.

Administrator response: No action required.

CTGKM2125E restore.TipadminPassword must not be fewer than 6 or more than 20 characters.

Explanation: restore.TipadminPassword cannot be shorter than 6 characters or longer than 20 characters.

System action: The operation fails.

Administrator response: Correct restore.TipadminPassword to a valid value.

CTGKM2126E The restore.ObfuscatedTipadminPassword parameter cannot be updated.

Explanation: The restore.ObfuscatedTipadminPassword parameter cannot be updated.

System action: The operation fails.

Administrator response: No action required.

CTGKM2201W Replication already in progress.

Explanation: Replication request rejected as one is already in progress.

System action: No action necessary.

Administrator response: Re-try replication when the currently running one has completed.

CTGKM2202E Replication failed for

Explanation: Replication has failed for the host listed.

System action: No action necessary.

Administrator response: No action necessary.

CTGKM2203E Replication failed with a connection error to

Explanation: Replication has failed for the host listed with a connection error.

System action: Check debug for exceptions.

Administrator response: Ensure that the hosts and ports listed are available.

CTGKM2204E Replication failed with a validation error to

Explanation: Replication has failed for the host listed with a validation error.

System action: Check debug for exceptions.

Administrator response: Check debug for exceptions.

CTGKM2206E IBM Security Key Lifecycle Manager Replication task has failed to start.

Explanation: IBM Security Key Lifecycle Manager Replication start command has failed.

System action: No action necessary.

Administrator response: See other error messages for further explanation.

CTGKM2207W IBM Security Key Lifecycle Manager Replication task is already up.

Explanation: IBM Security Key Lifecycle Manager Replication start command has been ignored as the task is already up.

System action: No action necessary.

Administrator response: No action necessary.

CTGKM2209E IBM Security Key Lifecycle Manager Replication task has failed to stop.

Explanation: IBM Security Key Lifecycle Manager Replication stop command has failed.

System action: No action necessary.

Administrator response: See other error messages for further explanation.

CTGKM2210W IBM Security Key Lifecycle Manager Replication task is already down.

Explanation: IBM Security Key Lifecycle Manager Replication stop command has been ignored as the task is already down.

System action: No action necessary.

Administrator response: No action necessary.

CTGKM2211E Command failed as the -confirm parameter is not set to Y.

Explanation: IBM Security Key Lifecycle Manager Replication stop command will not work without the confirm parameter being specified as Y.

System action: No action necessary.

Administrator response: If stop is required, rerun the command with the -confirm parameter set to Y.

CTGKM2212E Replication timed out to

Explanation: Replication for the specified host timed out.

System action: Check debug log.

Administrator response: Correct any problems on master or clone systems and retry.

CTGKM2213W Replication result unknown for

Explanation: Result for the replication of the specified host is unknown.

System action: Check debug log.

Administrator response: Check debug log.

CTGKM2214E Either both host name and port parameters must be coded, or neither.

Explanation: IBM Security Key Lifecycle Manager ReplicationNow command expects no parameters to replicate to all defined hosts, or both host name and port parameters to replicate to a single clone.

System action: No action necessary.

Administrator response: Re-run the command with correct parameters sepcified.

CTGKM2222E No valid replication config file exists.

Explanation: Either no replication config file exists, or it is invalid.

System action: No action necessary.

Administrator response: Create a valid replication config file prior to using IBM Security Key Lifecycle Manager replication commands.

CTGKM2237E Replication failed.

Explanation: Replication failed.

System action: No action necessary.

Administrator response: No action required.

CTGKM2243E Replication can only be invoked on the master machine.

Explanation: Replication now invoked from CLI on a clone machine. However, it can only in invoked on the master machine.

System action: No action necessary.

Administrator response: Go to master machine and invoke replication.

CTGKM2244E ***VALUE_0*** **can not be updated.**

Explanation: The config property referenced in the message can not be updated by the CLI. It is only updated by the product.

System action: No action necessary.

Administrator response: No action necessary.

CTGKM2245E Cannot modify the key

Explanation: Attempt to modify the key did not complete. The key that you intend to update, might not be found, or there might be a database error. There might be more information in the message that describes the problem.

System action: The key was not modified.

Administrator response: Additional information might guide your response. Make appropriate changes. Then, try the operation again.

CTGKM2300E Client certificate push is disabled.

Explanation: enableClientCertPush is set to false in the configuration file.

System action: The operation fails.

Administrator response: Change the value from false to true for the enableClientCertPush property in the SKLMConfig.properties file.

CTGKM2301E The number of pending client certificates has been reached.

Explanation: The number of pending client certificates exceeds the configuration value for maxPendingClientCerts.

System action: The operation fails.

Administrator response: Increase the value for the maxPendingClientCerts property in the SKLMConfig.properties file.

CTGKM2302E The X509 certificate cannot be null.

Explanation: The X509 certificate to be added to the pending client certificate list is null.

System action: The operation fails.

Administrator response: Add the X509 certificate and continue.

CTGKM2303E Client certificate exists in the pending client certificate list in the database: *VALUE_0*

Explanation: Client certificate exists in the pending client certificate list in the database.

System action: The operation fails.

Administrator response: None.

CTGKM2304E Client certificate alias is required.

Explanation: Client certificate alias is required.

System action: The operation fails.

Administrator response: Add the alias string and try again.

CTGKM2305E Client certificate UUID is required.

Explanation: The client certificate UUID is required.

System action: The operation fails.

Administrator response: Add the UUID and try again.

CTGKM2306E Client certificate alias already in use: *VALUE_0*

Explanation: The client certificate alias is already in use in the database and keystore.

System action: The operation fails.

Administrator response: Use a different alias and try again.

CTGKM2307E Client certificate UUID not found in the database: *VALUE_0*

Explanation: The client certificate UUID was not found in the database.

System action: The operation fails.

Administrator response: Change to a valid UUID and try again.

CTGKM2308E There are no pending client certificates in the database.

Explanation: There are no pending client certificates in the database.

System action: The operation fails.

Administrator response: No action.

CTGKM2311E maxPendingClientCerts does not fall within the valid value set of 1 to 999. *VALUE_0*

Explanation: maxPendingClientCerts does not fall within the valid value set of 1 to 999.

System action: The operation fails.

Administrator response: Update the configuration parameter and try again.

CTGKM2312E enableClientCertPush is not true or false.

Explanation: enableClientCertPush is not true or false.

System action: The operation fails.

Administrator response: Update the configuration parameter and try again.

CTGKM2313E maxPendingClientCerts does not fall within the valid value set of 1 to 999.

Explanation: maxPendingClientCerts does not fall within the valid value set of 1 to 999.

System action: The operation fails.

Administrator response: Update the configuration parameter and try again.

CTGKM2314E Certificate expired. Expired certificate cannot be used as client certificate.

Explanation: Certificate expired. Expired certificate cannot be used as client certificate.

System action: Update operation fails.

Administrator response: Specify a valid certificate alias and try the operation again.

CTGKM3010E Scheduler could not be found:\n Original error message: *error_message*

Explanation: Error in scheduler task detected.

System action: Operations involving scheduler will fail.

Administrator response: This should not happen under normal conditions. Investigate the logs and server startup log for clues. Restart the server.

CTGKM3020E Not a recognized device group: *VALUE_0*

Explanation: The specifed device group does not match any device group stored in the IBM Security Key Lifecycle Manager database.

System action: The operation fails.

Administrator response: Specify a valid device group and try again.

CTGKM3021E Operation cannot be null.

Explanation: The specifed KLMOperation object is null. This is an internal error.

System action: The operation fails.

Administrator response: This is an internal error that external users should not see. Contact IBM Support.

CTGKM3022E User *VALUE_0* **does not have appropriate permission to perform this operation. Depending on what the target resource is, it usually requires at least one of the listed permission(s):** *VALUE_1***.**

Explanation: The user does not have the appropriate permissions for this operation.

System action: The operation fails.

Administrator response: Check the user's permission. Refer to the IBM Security Key Lifecycle Manager information center to understand the permissions required for this operation. Set up appropriate permissions for the user and try again.

CTGKM3023E User *VALUE_0* **does not have a valid IBM Security Key Lifecycle Manager role. Some roles require both device group and action permissions. Verify that the user's role has appropriate permissions.**

Explanation: User does not have a valid IBM Security Key Lifecycle Manager role.

System action: The operation fails.

Administrator response: Check the user's permission. Refer to the IBM Security Key Lifecycle Manager information center to understand the permissions required for this operation. Set up appropriate permissions for the user and try again.

CTGKM3024E No permission set defined for operation *VALUE_0***.**

Explanation: No permission set defined for the specified operation.

System action: The operation fails.

Administrator response: This is an internal error that external users should not see. Contact IBM Support.

CTGKM3025E Operation *VALUE_0* requires device group permission, but specified resource is null.

Explanation: The specified operation requires device group permission but the device group resource is not specified.

System action: The operation fails.

Administrator response: The specified operation requires device group permission but the device group resource is not specified. This is an internal error that external users should not see. Contact IBM Support.

CTGKM3026E The target resource's device group name cannot be null.

Explanation: At permission checking, the target resource's device group name is not specified.

System action: This is an internal error that external users should not see. Contact IBM Support.

Administrator response: This is an internal error that external users should not see. Contact IBM Support.

CTGKM3027E Device group must be specified.

Explanation: Device group must be specified while invoking AuthorizationService.getPermission to get a user's IBM Security Key Lifecycle Manager permission.

System action: The operation fails.

Administrator response: The call to AuthorizationService.getPermission(KLMUserSession, String) must have a device group parameter specified. This is usually an internal error that external users should not see. If this API is invoked by another application, the application needs to adjust the parameter.

CTGKM3028E Cannot merge these two permissions.

Explanation: If two IBM Security Key Lifecycle Manager permissions have different device groups, they cannot be merged.

System action: The operation fails.

Administrator response: The call to KLMPermission.merge(KLMPermission) cannot merge the specified permission. This is usually an internal error that external users should not see. If this API is invoked by another application, the application needs to adjust the parameters.

CTGKM3029E No device group information specified.

Explanation: No device group information specified in the KLMDeviceType object.

System action: The operation fails.

Administrator response: KLMDeviceType object must include the device group name or device group ID. The caller needs to adjust the parameter.

CTGKM3030E User has no permission to query certificates. Check the user's permissions.

Explanation: Access is denied for query certificate operation. Check the user's permissions.

System action: The operation fails.

Administrator response: Check user's role and permissions. Assign appropriate permissions to the user. Then, try again.

CTGKM3031E User has no permission to query keys. Check the user's permissions.

Explanation: The permissions associated with the user role are not appropriate for query key operation.

System action: The operation fails.

Administrator response: Check the user's role and permissions. Assign appropriate permissions to the user. Then, try again.

CTGKM3032E User has no permission to query key groups. Check the user's permissions.

Explanation: The permissions associated with the user role are not appropriate for query key group operation.

System action: The operation fails.

Administrator response: Check the user's role and permissions. Assign appropriate permissions to the user. Then, try again.

CTGKM3033E User has no permission to query devices. Check user's permission.

Explanation: The permissions associated with the user role is not appropriate for query device operation.

System action: The device query fails.

Administrator response: Check user's role and permission set. Assign appropriate permissions to the user.

CTGKM3034E Cannot do 3592 rollover operation for non-3592 device group certificate.

Explanation: The certificate's device group must match the rollover device group.

System action: The operation fails.

Administrator response: Check the device group of the certificate and modify it to match the target rollover device group, or choose a different certificate for the rollover.

CTGKM3035E Cannot do LTO rollover operation for a non-LTO device group key group.

Explanation: The key group's device group must match the rollover device group.

System action: The operation fails.

Administrator response: Check the device group of the certificate and modify it to match the target rollover device group, or choose a different key group for the rollover.

CTGKM3036E Cannot find the specified rollover task.

Explanation: Cannot find the specified rollover task from the IBM Security Key Lifecycle Manager database.

System action: The operation fails.

Administrator response: Correct the specified rollover task parameters. Then, try again.

CTGKM3037E Failed to create key group *VALUE_0* **and keys with prefix** *VALUE_0*

Explanation: Failed to create the key group and keys with specified prefix.

System action: The operation fails.

Administrator response: Look at the logs for more information and retry. If the failure still exists, contact IBM support.

CTGKM3038E Cannot change the key in a DS5000 key group to another key group.

Explanation: DS5000 key group and keys are bound together. You cannot change the group membership of an individual DS5000 key directly.

System action: The operation fails.

Administrator response: You are not allowed to change the key of a DS5000 key group to another key group.

CTGKM3039E Cannot change the key's group membership. The key is used by one or more devices.

Explanation: Cannot change the key's group membership. The key is used by one or more devices.

System action: The operation fails.

Administrator response: You are not allowed to change the key of a DS5000 key group to another key group. The key is used by one or more devices.

CTGKM3040E Object with identifier *object_id* **cannot be found.**

Explanation: The identifier value that you specified does not match an existing object.

System action: The operation fails.

Administrator response: Specify an identifier that corresponds to an existing object.

CTGKM3041E User *object_id* **has no access to the destroyed object. The destroyed object has no device group information and only the klmSecurityOfficer role can access it.**

Explanation: When a KLM key or certificate is marked as destroyed, its data in the relation table are removed. There is no device group information. Only the klmSecurityOfficer role can access the object.

System action: The operation fails.

Administrator response: Log in as security officer to view the destroyed objects.

CTGKM3042E The alias of the key that encrypts the secret key file must be specified.

Explanation: To import the secret key file, specify the alias of the public private key pair so that IBM Security Key Lifecycle Manager can get the private key to decrypt the file.

System action: The operation fails.

Administrator response: Specify the keyAlias when you run the tklmKeyImport command.

CTGKM3043E Error occurred while loading data from the file *VALUE_0***. Make sure that the password is correct and the file has not been tampered with.**

Explanation: This is an error in reading data from a key or certificate file.

System action: The operation fails.

Administrator response: Ensure that the path and filename are correct, and that the password is correct. Then, try the operation again.

CTGKM3044E The file *VALUE_0* **is reserved for internal IBM Security Key Lifecycle Manager keystore. Use another file name.**

Explanation: User cannot create a new keystore that has the same location and file name as the internal IBM Security Key Lifecycle Manager keystore.

System action: The operation fails.

Administrator response: Use a different file name. Then, try the operation again.

CTGKM3045E Value for configuration parameter Audit.eventQueue.max is not valid. A valid value is a non-negative integer.

Explanation: Audit.eventQueue.max is a non-negative integer.

System action: The operation fails.

Administrator response: Specify 0 or a positive integer and try the operation again.

CTGKM3046E Value for configuration parameter Audit.handler.file.size is not valid. A valid value is a non-negative integer.

Explanation: Audit.handler.file.size is a non-negative integer.

System action: The operation fails.

Administrator response: Specify 0 or a positive integer and try the operation again.

CTGKM3047E Value for configuration parameter Audit.handler.file.threadlifespan is not valid. A valid value is a non-negative integer.

Explanation: Audit.handler.file.threadlifespan is a non-negative integer.

System action: The operation fails.

Administrator response: Specify 0 or a positive integer and try the operation again.

CTGKM3048E Value for configuration parameter Audit.handler.file.multithreads is not valid. A valid value is true or false.

Explanation: Audit.handler.file.multithreads parameter allows the use of multiple threads while logging audit events. The valid value is either true or false.

System action: The operation fails.

Administrator response: Specify true or false and try the operation again.

CTGKM3049E Configuration parameter tklm.backup.dir cannot be modified directly. To change to a different directory, specify the new directory when doing the next backup.

Explanation: You cannot modify the current backup directory. To change to a different directory, specify the new directory when doing the next backup.

System action: The operation fails.

Administrator response: Do not use the tklmConfigUpdateEntry command to modify the tklm.backup.dir configuration parameter.

CTGKM3050E Value for configuration parameter cert.valiDATE is not valid. A valid value is true or false.

Explanation: Valid value for cert.valiDATE is true or false.

System action: The operation fails.

Administrator response: Specify true or false as the value.

CTGKM3051E Keystore name cannot be modified by updating configuration parameter config.keystore.name directly. Use a keystore command to update the name.

Explanation: Keystore name cannot be modified by updating config.keystore.name directly. Use a keystore command to update the name.

System action: The operation fails.

Administrator response: Keystore name cannot be modified by updating config.keystore.name directly. Use a keystore command to update the name.

CTGKM3052E Invlalid value for configuration parameter disableDatabaseBackup. A valid value is true or false.

Explanation: Value for configuration parameter disableDatabaseBackup is not valid. A valid value is true or false.

System action: The operation fails.

Administrator response: Specify true or false as the input value.

CTGKM3053E Value for configuration parameter fips is not valid. A valid value is on or off.

Explanation: Value for configuration parameter fips is not valid. A valid value is on or off.

System action: The operation fails.

Administrator response: Specify on or off as the input value.

CTGKM3054E Value for configuration parameter requireHardwareProtectionForSymmetricKeys is not valid. A valid value is true or false.

Explanation: Value for configuration parameter requireHardwareProtectionForSymmetricKeys is not valid. A valid value is true or false.

System action: The operation fails.

Administrator response: Specify true or false as the input value.

CTGKM3055E Configuration parameter tklm.backup.db2.dir cannot be modified directly. To change to a different directory, specify the new directory when doing the next backup.

Explanation: You cannot modify the current backup directory. To change to a different directory, specify the new directory when doing the next backup.

System action: The operation fails.

Administrator response: Do not use the tklmConfigUpdateEntry command to modify tklm.backup.db2.dir.

CTGKM3056E Value for configuration parameter tklm.encryption.pbe.algorithm is not valid. A valid value is PBEWithMD5AndTripleDES.

Explanation: Value for configuration parameter tklm.encryption.pbe.algorithm is not valid. A valid value is PBEWithMD5AndTripleDES.

System action: The operation fails.

Administrator response: Specify PBEWithMD5AndTripleDES as the input value.

CTGKM3057E Value for configuration parameter TransportListener.ssl.clientauthentication is not valid. A valid value is 0, 1 or 2.

Explanation: Value for configuration parameter TransportListener.ssl.clientauthentication is not valid. A valid value is 0, 1 or 2.

System action: The operation fails.

Administrator response: Specify 0, 1 or 2 as the input value.

CTGKM3058E Value for configuration parameter TransportListener.ssl.port is not valid. A valid value is an integer between 1 and 65535.

Explanation: Value for configuration parameter TransportListener.ssl.port is not valid. A valid value is an integer between 1 and 65535.

System action: The operation fails.

Administrator response: Specify an integer between 1 and 65535 as the input value.

CTGKM3059E Value for configuration parameter TransportListener.ssl.protocols is not valid. A valid value is SSL_TLS, SSL, TLS, or SSL_TLSv2.

Explanation: Value for configuration parameter TransportListener.ssl.protocols is not valid. A valid value is SSL_TLS, SSL, TLS, or SSL_TLSv2.

System action: The operation fails.

Administrator response: Specify SSL_TLS, SSL, TLS, or SSL_TLSv2 as the input value.

CTGKM3060E Value for configuration parameter TransportListener.ssl.timeout is not valid. A valid value is an integer between 1 and 120.

Explanation: Value for configuration parameter TransportListener.ssl.timeout is not valid. A valid value is an integer between 1 and 120.

System action: The operation fails.

Administrator response: Specify an integer between 1 and 120 as the input value.

CTGKM3061E Value for configuration parameter TransportListener.tcp.port is not valid. A valid value is an integer between 1 and 65535.

Explanation: Value for configuration parameter TransportListener.tcp.port is not valid. A valid value is an integer between 1 and 65535.

System action: The operation fails.

Administrator response: Specify an integer between 1 and 65535 as the input value and ensure that the port is not used by other applications on the system.

CTGKM3062E Value for configuration parameter TransportListener.tcp.timeout is not valid. A valid value is an integer between 1 and 120.

Explanation: Value for configuration parameter TransportListener.tcp.timeout is not valid. A valid value is an integer between 1 and 120.

System action: The operation fails.

Administrator response: Specify an integer between 1 and 120 as the input value.

CTGKM3063E Value for configuration parameter stopRoundRobinKeyGrps is not valid. A valid value is true or false.

Explanation: Value for configuration parameter stopRoundRobinKeyGrps is not valid. A valid value is true or false.

System action: The operation fails.

Administrator response: Specify true or false.

CTGKM3064E Value for configuration parameter useSKIDefaultLabels is not valid. A valid value is true or false.

Explanation: Value for configuration parameter useSKIDefaultLabels is not valid. A valid value is true or false.

System action: The operation fails.

Administrator response: Specify true or false.

CTGKM3065E Value for configuration parameter zOSCompatibility is not valid. A valid value is true or false.

Explanation: Value for configuration parameter zOSCompatibility is not valid. A valid value is true or false.

System action: The operation fails.

Administrator response: Specify true or false.

CTGKM3066E Value for configuration parameter pcache.refresh.interval is not valid. A valid value is a positive integer.

Explanation: Value for configuration parameter pcache.refresh.interval is not valid. A valid value is a positive integer.

System action: The operation fails.

Administrator response: Specify a positive integer.

CTGKM3067E Failed to create the directory for the log file:

Explanation: Cannot create a new directory as specified.

System action: The operation fails.

Administrator response: Specify a valid directory name and try again.

CTGKM3068E Cannot create the new log file: *VALUE_0*

Explanation: Cannot create a new log file as specified.

System action: The operation fails.

Administrator response: Specify a valid file name and try again.

CTGKM3069E Not a valid file name. Specify a path and file name that is relative to SKLM_HOME.

Explanation: Not a valid file name. Specify a path and file name that is relative to SKLM_HOME.

System action: The operation fails.

Administrator response: Specify a valid file and path name and try again.

CTGKM3071E Cannot set the certificate alias because the keystore is not defined.

Explanation: There is no certificate alias to be set because the keystore is not defined.

System action: The operation fails.

Administrator response: Add or create a keystore for IBM Security Key Lifecycle Manager. Specify the certificate alias that exists in the keystore and try again.

CTGKM3072E The specified certificate has a different usage. Use a different certificate.

Explanation: The specified certificate has a different usage. Use a different certificate.

System action: The operation fails.

Administrator response: Specify a different certificate alias and try again.

CTGKM3073E Cannot find the class in classpath: *VALUE_0*

Explanation: Cannot find the class in classpath.

System action: The operation fails.

Administrator response: Make sure the class and package name are correct in the configuration file.

CTGKM3074E Not a valid class name. The class object cannot be instantiated: *VALUE_0*

Explanation: Not a valid class name. The class object cannot be instantiated.

System action: The operation fails.

Administrator response: Make sure the class and package name are correct in the configuration file.

CTGKM3075E Not a valid class name. It must implement SecurityEventHandlerSpi class: *VALUE_0*

Explanation: Not a valid class name. It must implement SecurityEventHandlerSpi class

System action: The operation fails.

Administrator response: Make sure the class and package name are correct in the configuration file.

CTGKM3076E Unsupported event type: *VALUE_0*

Explanation: Unsupported event type.

System action: The operation fails.

Administrator response: Make sure the event type specified in the configuration file is correct.

CTGKM3077E Value for configuration parameter *VALUE_0* **is not valid. Valid values are success, failure, or both that are separated by comma or semicolon.**

Explanation: Valid values are success, failure, or both that are separated by a comma or semicolon.

System action: The operation fails.

Administrator response: Make sure the value is success, failure, or both that are separated by a comma or semicolon.

CTGKM3078E Keystore is not defined. The TransportListener.ssl.ciphersuites parameter cannot be set to the specified value.

Explanation: Keystore is not defined. The TransportListener.ssl.ciphersuites parameter cannot be set to the specified value.

System action: The operation fails.

Administrator response: Create the keystore and certificates and then try the operation again.

CTGKM3079E TrustManager and KeyManager are not initialized. The TransportListener.ssl.ciphersuites parameter cannot be set to the specified value.

Explanation: TrustManager and KeyManager are not initialized. The TransportListener.ssl.ciphersuites parameter cannot be set to the specified value.

System action: The operation fails.

Administrator response: Create the keystore and certificates and then try the operation again.

CTGKM3080E SSLContext is not initialized. The TransportListener.ssl.ciphersuites parameter cannot be set to the specified value.

Explanation: SSLContext is not initialized. The TransportListener.ssl.ciphersuites parameter cannot be set to the specified value.

System action: The operation fails.

Administrator response: Create the keystore and certificates and then try the operation again.

CTGKM3081E Not a supported cipher suite: *VALUE_0*

Explanation: Not a supported cipher suite.

System action: The operation fails.

Administrator response: Specify a different value and try the operation again.

CTGKM3082E No SSL certificate alias defined in the configuration file. TrustManager and KeyManager are not initialized. The TransportListener.ssl.ciphersuites parameter cannot be set to the specified value.

Explanation: No SSL certificate alias defined in the configuration file. TrustManager and KeyManager are not initialized. The TransportListener.ssl.ciphersuites parameter cannot be set to the specified value.

System action: The operation fails.

Administrator response: Set config.keystore.ssl.certalias in the configuration file and try the operation again.

CTGKM3083E Configuration parameter tklm.encryption.password cannot be updated.

Explanation: Configuration parameter tklm.encryption.password cannot be updated.

System action: The operation fails.

Administrator response: Configuration parameter tklm.encryption.password cannot be updated.

CTGKM3084E Value for configuration parameter numberOfKeys is not valid. A valid value is a positive integer

Explanation: Value for configuration parameter numberOfKeys is not valid. A valid value is a positive integer, such as 12.

System action: The operation fails.

Administrator response: Specify a positive integer.

CTGKM3085E To export the secret key, user must have proper permissions on the certificate and public key that are used to encrypt the secret key file.

Explanation: To export the secret key, user must have proper permissions on the certificate and public key that are used to encrypt the secret key file.

System action: The operation fails.

Administrator response: Give the user the proper permissions on the certificate and try again.

CTGKM3086E To import the secret key, user must have the proper permissions on the private key that is used to decrypt the secret key file.

Explanation: To import the secret key, user must have the proper permissions on the private key that is used to decrypt the secret key file.

System action: The operation fails.

Administrator response: Give the user the proper permissions on the private key and try again.

CTGKM3087E Deletion is not permitted on the object. Check the enableKMIPDelete flag on the object's device group.

Explanation: Deletion is not permitted on the object when the enableKMIPDelete flag is turned off.

System action: The operation fails.

Administrator response: Turn on the enableKMIPDelete flag on the object's device group and try the operation again.

CTGKM3088E Not a recognized device group internal identifier: *VALUE_0*

Explanation: The specifed device group internal identifier does not match any device group stored in the IBM Security Key Lifecycle Manager database.

System action: The operation fails.

Administrator response: Contact IBM Support.

CTGKM3089E Credential is not specified in KMIPUserSession.

Explanation: KMIPUserSession object must provide the user credential for authorization purpose

System action: The authorization of the KMIP user failed.

Administrator response: Check if the KMIP client provides an appropriate credential such as correct client certificate.

CTGKM3090E The KMIP user is not authorized to access the target object.

Explanation: The KMIP user is not authorized to access the target object.

System action: The authorization of the KMIP user failed.

Administrator response: Check if the KMIP client provides appropriate credentials such as the correct client certificate.

CTGKM3091E There is no KMIP policy defined for the operation.

Explanation: There is no KMIP policy defined for the operation.

System action: The authorization of the KMIP user failed.

Administrator response: The requested KMIP operation may not be supported. Contact IBM Support.

CTGKM3092E KLMResource is not properly instantiated.

Explanation: KLMResource is not properly instantiated.

System action: The authorization of the KMIP user failed.

Administrator response: Contact IBM Support.

CTGKM3093E Device group must be specified as a KMIP user credential.

Explanation: Device group must be specified as a KMIP user credential.

System action: The authorization of the KMIP user failed.

Administrator response: Check if the KMIP request message includes device metadata information.

CTGKM3094E Name or ID of a device group must be specified in device metadata as a KMIP user credential.

Explanation: Name or ID of a device group must be specified in device metadata as a KMIP user credential.

System action: The authorization of the KMIP user failed.

Administrator response: Check if the KMIP request message includes a name or ID of a device group as part of device metadata information.

CTGKM3095E KMIP user's device metadata credential does not match device group information in the target resource.

Explanation: KMIP user's device metadata credential does not match device group information in the target resource.

System action: The authorization of the KMIP user failed.

Administrator response: Contact IBM Support for assistance.

CTGKM3096E Credential in KMIP user session is not properly specified.

Explanation: Credential in KMIP user session is not properly specified.

System action: The authorization of the KMIP user failed.

Administrator response: Contact IBM Support for assistance.

CTGKM3097E User has no authority to access the target object. Access requires action permission on device group, klmConfigure or klmSecurityOfficer permission.

Explanation: User has no authority to access the target object. Access requires action permission on device group, klmConfigure or klmSecurityOfficer permission.

System action: Authorization fails.

Administrator response: Check if the user has appropriate permissions.

CTGKM3098E Certificate expired. Expired certificate cannot be used as default SSL or IKEv2-SCSI certificate.

Explanation: Certificate expired. An expired certificate cannot be used as a default SSL or IKEv2-SCSI certificate.

System action: Update operation fails.

Administrator response: Specify a valid certificate alias and try the operation again.

CTGKM3099E Keystore name *VALUE_0* **for keystore UUID** *VALUE_1* **is not valid.**

Explanation: The specified keystore name is not valid.

System action: The keystore list operation fails.

Administrator response: Specify a different keystore name and try the operation again.

CTGKM3100E Value for configuration parameter lock.timeout is not valid. A valid value is a non negative integer.

Explanation: The specified lock.timeout value is not valid.

System action: The operation to update the lock.tmeout value failed.

Administrator response: Specify a different value and try the operation again.

CTGKM3101E The operation failed as dependent data has been locked. Another user might be accessing the same data. Try the operation again later.

Explanation: The dependant data has been locked and the current thread failed to acquire the lock on the data within limited time frame. It may be because another user is accessing the same data.

System action: Try the operation again.

Administrator response: Try the operation again.

CTGKM3102E *VALUE_0* **and** *VALUE_1* **cannot use the same port.**

Explanation: TCP, KMIP, SSL cannot use the same port.

System action: The operation fails.

Administrator response: Try the operation again.

CTGKM3103E Value for configuration parameter KMIPListener.ssl.port is not valid. A valid value is an integer between 1 and 65535.

Explanation: Value for configuration parameter KMIPListener.ssl.port is not valid. A valid value is an integer between 1 and 65535.

System action: The operation fails.

Administrator response: Specify an integer between 1 and 65535 as the input value and ensure that the port is not used by other applications on the system.

CTGKM3104E Value for configuration parameter backup.keycert.before.serving is not valid. The value must be either true or false.

Explanation: Value for configuration parameter backup.keycert.before.serving is not valid. The value must be either true or false.

System action: The operation fails.

Administrator response: Specify either true or false as the value.

CTGKM3105E Cannot use the certificate for key export operation. The certificate contains an EC key.

Explanation: The certificate contains an EC public key. The IBM JVM 5.0 does not support key encryption using EC key.

System action: The operation fails. Specify a different certificate and try the operation again.

Administrator response: Specify a different certificate and try the operation again.

CTGKM3106E Key or certificate with alias or key prefix *VALUE_0* was not served because it is not backed up.

Explanation: Keys or certificates must be backed up before they can be served.

System action: The operation fails.

Administrator response: First back up IBM Security Key Lifecycle Manager. Then, try the operation again.

CTGKM3107E Value for configuration parameter autoRestartAfterRestore is not valid. The value must be either true or false.

Explanation: Value for configuration parameter autoRestartAfterRestore is not valid. The value must be either true or false.

System action: The operation fails.

Administrator response: Specify either true or false as the value.

CTGKM3109E Key or certificate with alias or key prefix *VALUE_0* was not served because it has not been released.

Explanation: Keys or certificates must be released before they can be served. This message may also occur if the config property release.date is missing or wrongly formatted.

System action: The operation fails.

Administrator response: First, run the command tklmKeyRelease. Then try the operation again.

CTGKM3110E Value for configuration parameter enableKeyRelease is not valid. The value must be either true or false.

Explanation: Value for configuration parameter enableKeyRelease is not valid. The value must be either true or false.

System action: The operation fails.

Administrator response: Specify either true or false as the value.

CTGKM3111E Value for configuration parameter requireSHA2Signatures is not valid. The value must be either true or false.

Explanation: Value for configuration parameter requireSHA2Signatures is not valid. The value must be either true or false.

System action: The operation fails.

Administrator response: Specify either true or false as the value.

CTGKM4000E The configuration property pkcs11.pin.obfuscated can not be updated.

Explanation: The configuration property pkcs11.pin.obfuscated is set by the product and can not be updated by a user.

System action: Update operation fails.

Administrator response: Specify a configuration paramter that can be updated and try again.

CTGKM4001E Specified PKCS 11 configuration path or filename does not exist.

Explanation: The path and/or file name specified does not exist.

System action: Update operation fails.

Administrator response: Specify a valid path and file name for the PKCS 11 configuration file.

CTGKM6002E Bad Request: Invalid user authentication ID or invalid request format.

Explanation: An incorrect request format or user ID was used for authentication.

System action: Request fails.

Administrator response: Specify a correct request format and a valid user ID.

CTGKM6003E Authentication Failure: Incorrect user ID or password.

Explanation: Incorrect user id or password specified.

System action: Login fails.

Administrator response: Specify correct user ID or password.

CTGKM6004E User is not authenticated or has already logged out.

Explanation: User is not authenticated or has already logged out.

System action: Login fails.

Administrator response: Specify correct user id or password.

CTGKM6027E key group entry must specify either entry uuid, or alias.

Explanation: Either uuid or alias should be provided for key group entry.

System action: Key group entry add fails.

Administrator response: Specify either uuid or alias.

CTGKM6060E Previous run not completed. Must use the same master key size - *KEY_SIZE*

Explanation: Processing failed when creating new master key for encryption. A previous run of this process was not completed. Must use the same master key size.

System action: The operation fails.

Administrator response: Retry the operation.

CTGKM6061E Unable to read/initialize status: *ERROR_MESSAGE*

Explanation: Processing failed when creating new master key for encryption. Unable to read/initialize status.

System action: The operation fails.

Administrator response: Retry the operation.

CTGKM6062E Unable to create Master Key: MasterKey = null

Explanation: Processing failed when creating new master key for encryption. Unable to create Master Key: MasterKey = null

System action: The operation fails.

Administrator response: Retry the operation.

CTGKM6063E Error creating Master Key: *ERROR_MESSAGE*

Explanation: Processing failed when creating new master key for encryption. Error creating Master Key.

System action: The operation fails.

Administrator response: Retry the operation.

CTGKM6064E Error when encrypting key DB Table data with new MasterKey: *ERROR_MESSAGE*

Explanation: Processing failed when creating new master key for encryption. Error when encrypting key DB Table data with new MasterKey.

System action: The operation fails.

Administrator response: Retry the operation.

CTGKM6065E Error when encrypting certificate DB Table data with new MasterKey: *ERROR_MESSAGE*

Explanation: Processing failed when creating new master key for encryption. Error when encrypting certificate DB Table data with new MasterKey.

System action: The operation fails.

Administrator response: Retry the operation.

CTGKM6066E Error when encrypting SecretData DB Table data with new MasterKey: *ERROR_MESSAGE*

Explanation: Processing failed when creating new master key for encryption. Error when encrypting SecretData DB Table data with new MasterKey.

System action: The operation fails.

Administrator response: Retry the operation.

CTGKM6067E Error when updating MasterKey size configuration: *ERROR_MESSAGE*

Explanation: Processing failed when creating new master key for encryption. Error when updating MasterKey size configuration.

System action: The operation fails.

Administrator response: Retry the operation.

CTGKM6068E Error when updating SKLM KeyStore with the new MasterKey: *ERROR_MESSAGE*

Explanation: Processing failed when creating new master key for encryption. Error when updating SKLM KeyStore with the new MasterKey.

System action: The operation fails.

Administrator response: Retry the operation.

CTGKM7000E Either both Audit.syslog.server.host and Audit.syslog.server.port configuration parameters must be provided, or neither. When provided the value of Audit.isSyslog configuration parameter should be true.

Explanation: Either both Audit.syslog.server.host and Audit.syslog.server.port configuration parameters must be provided, or neither. When provided the value of Audit.isSyslog configuration parameter should be true.

System action: Login fails.

Administrator response: Either both Audit.syslog.server.host and Audit.syslog.server.port configuration parameters must be provided, or neither. When provided the value of Audit.isSyslog configuration parameter should be true.

CTGKM7001E Value for configuration parameter Audit.syslog.server.port is not valid. A valid value is an integer between 1 and 65535.

Explanation: Value for configuration parameter Audit.syslog.server.port is not valid. A valid value is an integer between 1 and 65535.

System action: The operation fails.

Administrator response: Specify an integer between 1 and 65535 as the input value.

CTGKM7002E Value for configuration parameter Audit.syslog.server.host is not valid. It should not be greater than 255 characters.

Explanation: Value for configuration parameter Audit.syslog.server.host is not valid. It should not be greater than 255 characters.

System action: The operation fails.

Administrator response: Specify a valid value for host as the input value.

CTGKM7003E Value for configuration parameter Audit.isSyslog is not valid. The value must be either true or false.

Explanation: Value for configuration parameter Audit.isSyslog is not valid. The value must be either true or false.

System action: The operation fails.

Administrator response: Specify either true or false as the input value.

CTGKM7004E Unable to connect to Server to log Audit events.

Explanation: Unable to connect to Server to log Audit events.

System action: The operation fails.

Administrator response: Please check values of configuration parameters and check whether the server is running on the specific port.

CTGKM7005E Value for configuration parameter Audit.syslog.isSSL is not valid. The value must be either true or false.

Explanation: Value for configuration parameter Audit.syslog.isSSL is not valid. The value must be either true or false.

System action: The operation fails.

Administrator response: Specify either true or false as the input value.

CTGKN1000W The SSL/KMIP or IKEv2-SCSI certificate configuration cannot be updated before a master keystore has been created. Click on the master keystore link to create it now. Select OK to update Audit and Key Serving Parameters or Ports information.

Explanation: A master keystore must exist before you can define an SSL/KMIP or IKEv2-SCSI server certificate. Before you create the keystore, you can specify the audit level or other configuration settings.

System action: No server certificates are specified.

Administrator response: First, create the master keystore. Then define the SSL/KMIP or IKEv2-SCSI server certificate. Otherwise, you can specify the audit level or other configuration settings.

CTGKN1003E File name entered for the certificate to be imported is not valid. Double-click on the ${0} column of the selected certificate entry to enter a valid path and file name.

Explanation: The file name that you entered is not a valid file name.

System action: The file name does not exist in the specified path.

Administrator response: Specify a correct, existing path and file name. Then try again.

CTGKN1005W This panel cannot be accessed before a master keystore has been created. Click on the master keystore link to create it now.

Explanation: First, you must create a master keystore.

System action: The content of the page is not displayed until a keystore is created.

Administrator response: Create the master keystore. Then try again.

CTGKN1006I Key name specified is not a known key.

Explanation: The key that you specified is unknown.

System action: The operation fails.

Administrator response: Obtain a valid key name. Then try again.

CTGKN1007E Insufficient permission to import the certificate.

Explanation: Your role must have a permission to the create action and a permission to the appropriate device group. Or, your role must have a permission to the configure action to import an SSL or KMIP, certificate.

System action: The import operation fails.

Administrator response: Obtain a user ID with the required permissions. Then try again.

CTGKN1008E Insufficient permission to add additional keys to the key group.

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

System action: The operation fails.

Administrator response: Obtain a user ID with the permissions required to add keys to a key group. Then try again.

CTGKN1011W You cannot go to the Key and Device Management panel because the certificate belongs to an ${0} device group.

Explanation: Because the certificate belongs to an UNKNOWN or CONFLICTED device group, the system cannot display a specific key and device management panel for a device group.

System action: The system issues this warning.

Administrator response: Click on the Key and Device Management navigation link and select the panel for the device group that you want to use this certificate.

CTGKN1012W Are you sure you would like to reject the certificate with a subject distinguished name of ${0} and issuer distinguished name of ${1}?

Explanation: The message confirms that you want to reject the selected client device communication certificate that was pushed to the IBM Security Key Lifecycle Manager server from a device.

System action: Confirming the message will remove the certificate from IBM Security Key Lifecycle Manager. The certificate will not be able to be used for secure communications between the device and the server. The certificate will not be added to the keystore.

Administrator response: Ensure that the subject distinguished name and issuer distinguished name correctly identify the client device communication certificate that you intend to reject. Then click **OK** to remove the certificate. For more information to identify the certificate before rejection, click **Cancel** and then select **View** for the certificate.

CTGKN1014E The selected certificate has expired and therefore cannot be accepted. Select Reject to remove it from the table.

Explanation: A client device communication certificate that was pushed to the IBM Security Key Lifecycle Manager server from a device has expired before being accepted. Expired certificates cannot be accepted.

System action: The certificate is not added to the IBM Security Key Lifecycle Manager keystore.

Administrator response: Select **Reject** to remove the certificate from the list of pending client device communication certificates.

CTGKO0000E Password policy authority error occurred: *VALUE_0*

Explanation: Unspecified general error occurred.

System action: Cannot process the password policy authority request.

Administrator response: Contact your administrator.

CTGKO0002E Cannot retrieve password policy from data store - *VALUE_0*. **Cause:** *VALUE_1*

Explanation: Password policy could not be retrieved.

System action: Cannot process the password policy authority request.

Administrator response: Contact an administrator who has read and write access to the TKLMPasswordPolicy.xml file. This is usually the Tivoli Integrated Portal administrator (TIPAdmin) or Security Key Lifecycle Manager administrator (SKLMAdmin). Only the TIPAdmin user can add or modify user profiles and test the password policy.

CTGKO0003E Cannot read the retrieved password policy - *VALUE_0*. **Cause:** *VALUE_1*

Explanation: Cannot parse the password policy definition. The data might be corrupt.

System action: Cannot process the password policy authority request.

Administrator response: Contact an administrator who has read and write access to the TKLMPasswordPolicy.xml file. This is usually the Tivoli Integrated Portal administrator (TIPAdmin) or Security Key Lifecycle Manager administrator (SKLMAdmin). Only the TIPAdmin user can add or modify user profiles and test the password policy.

CTGKO0004E Cannot remove password policy from data store - *VALUE_0*. **Cause:** *VALUE_1*

Explanation: The password policy is not removed from the data store.

System action: Cannot process the password policy authority request.

Administrator response: Contact an administrator who has read and write access to the TKLMPasswordPolicy.xml file. This is usually the Tivoli Integrated Portal administrator (TIPAdmin) or Security Key Lifecycle Manager administrator (SKLMAdmin). Only the TIPAdmin user can add or modify user profiles and test the password policy.

CTGKO0005E Cannot determine password policy location - *VALUE_0*. **Cause:** *VALUE_1*

Explanation: Password policy location can not be determined. Check your product configuration

System action: Cannot process the password policy authority request.

Administrator response: Contact an administrator who has read and write access to the TKLMPasswordPolicy.xml file. This is usually the Tivoli Integrated Portal administrator (TIPAdmin) or Security Key Lifecycle Manager administrator (SKLMAdmin). Only the TIPAdmin user can add or modify user profiles and test the password policy.

CTGKO0100E Password policy violation was detected. Password is too long. Maximum length is *VALUE_0*.

Explanation: The password length cannot exceed the value set in the password policy.

System action: No password is set. A new user profile is not created.

Administrator response: Submit a password not greater than the maximum allowed length. Retry the request or contact your administrator.

CTGKO0101E Password policy violation was detected. Password is too short. Minimum length is *VALUE_0*.

Explanation: The password length cannot be less than the value set in the password policy.

System action: No password is set. A new user profile is not created.

Administrator response: Submit a password that has a valid minimum length. Retry the request or contact your administrator.

CTGKO0102E Password policy violation was detected. Password does not contain a required character. One of the following characters is required: *VALUE_0*.

Explanation: The password must contain a required character specified in the password policy.

System action: No password is set. A new user profile is not created.

Administrator response: Submit a password containing one of the required characters. Retry the request or contact your administrator.

CTGKO0103E Password policy violation was detected. Password contains an incorrect character. Any of the following characters may not be used: *VALUE_0*.

Explanation: The password cannot contain characters that password policy specifies should not be used.

System action: No password is set. A new user profile is not created.

Administrator response: Submit a password without incorrect characters. Retry the request or contact your administrator.

CTGKO0104E Password policy violation was detected. Password contains too many consecutive occurrences of the same character. Maximum number of occurrences is: *VALUE_0*.

Explanation: The password cannot contain more than the maximum consecutive occurrences of the same character.

System action: No password is set. A new user profile is not created.

Administrator response: Submit a password that does not exceed the maximum consecutive occurrences of the same character. Retry the request or contact your administrator.

CTGKO0105E Password policy violation was detected. Password does not contain the minimum required number of numeric characters. Minimum number of numeric characters is: *VALUE_0*.

Explanation: The password must contain a minimum number of numeric characters.

System action: No password is set. A new user profile is not created.

Administrator response: Submit a password containing at least the required minimum number of numeric characters. Retry the request or contact your administrator.

CTGKO0106E Password policy violation was detected. Password does not contain the minimum required number of alphabetic characters. Minimum number of alphabetic characters is: *VALUE_0*.

Explanation: The password must contain a minimum number of alphabetic characters.

System action: No password is set. A new user profile is not created.

Administrator response: Submit a password containing at least the required minimum number of alphabetic characters. Retry the request or contact your administrator.

CTGKO0107E Password policy violation was detected. Password does not contain the minimum required number of unique characters. Minimum number of unique characters is: *VALUE_0*.

Explanation: The password must contain a minimum number of unique characters.

System action: No password is set. A new user profile is not created.

Administrator response: Submit a password containing at least the minimum number of unique characters specified by the password policy. Retry the request or contact your administrator.

CTGKO0108E Password policy violation was detected. Password contains the user ID.

Explanation: Password policy violation was detected.

System action: No password is set. A new user profile is not created.

Administrator response: Submit a password that does not contain the user ID. Retry the request or contact your administrator.

CTGKO0109E Password policy violation was detected. Password contains user name.

Explanation: A password cannot contain any part of the user name.

System action: No password is set. A new user profile is not created.

Administrator response: Submit a password that does not contain any part of the user name. Retry the request or contact your administrator.

CTGKP5001E Unable to decode attribute value.

Explanation: Cannot process a message. There is an error on the input received in the message.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP5002E Index may not be specified in Add Attribute operation.

Explanation: Cannot process a message. There is an error on the input received in the message.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP5003E Service returned no KMIPCryptographicObject.

Explanation: Internal error. After processing the message, KMIP service did not return KMIPCryptographicObject.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP5004E Unique Identifier mismatch : query does not match response.

Explanation: Internal error. After processing the message, Unique Identifier in the query and response do not match.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP5005E Single valued attribute *VALUE_0* **is already present.**

Explanation: Cannot process the message. Received multiple values for a single valued attribute. Expected only one value.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP5006E A non-zero index passed for a single valued attribute *VALUE_0*.

Explanation: Cannot process the message. Index for the single valued attribute must to be zero.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP5007E A non-typed Digest appeared as an attribute.

Explanation: Cannot process the message. There is an error on the input received in the message.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP5008E A non-string value appeared in a custom attribute.

Explanation: Cannot process the message, incorrect value for custom attribute received. All custom attributes must be of type TEXT_STRING.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP5009E Unknown attribute name *VALUE_0*.

Explanation: Cannot process the message, unrecognized attribute name.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP5010E A non-typed Name appeared as an attribute.

Explanation: Cannot process the message, unrecognized type for Name attribute.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP5011E KMIP SSL Listener is up.

Explanation: IBM Security Key Lifecycle Manager is ready to accept KMIP requests from a client.

System action: IBM Security Key Lifecycle Manager is ready to accept KMIP requests from a client.

Administrator response: None.

CTGKP5012E Cannot obtain IBM Security Key Lifecycle Manager keystore password.

Explanation: Cannot process a message, internal error occurred.

System action: The requested operation fails.

Administrator response: Check the audit logs. Correct the input and retry the operation.

CTGKP5013E Trustmanagers or keymanagers cannot be initialized.

Explanation: Internal error occurred, could not complete initialization for KMIP SSL Listener. Make sure KMIP is configured correctly for SSL.

System action: IBM Security Key Lifecycle Manager is not ready to accept KMIP requests from a client.

Administrator response: Check the audit logs. Correct the input and retry the operation.

CTGKP5014E KMIP SSL Listener did not come up.

Explanation: Error occurred getting KMIP SSL Listener up. IBM Security Key Lifecycle Manager is not ready to accept KMIP requests from a client.

System action: Cannot accept KMIP messages.

Administrator response: Check the audit logs. Correct the input and retry the operation.

CTGKP5015E The *VALUE_0* attribute requires a non-null value.

Explanation: The attribute can not contain a null value.

System action: Will not process the message.

Administrator response: Correct the input and retry the message.

CTGKS0001E Keystore password for *VALUE_0* is null.

Explanation: The keystore is not found in the IBM Security Key Lifecycle Manager database. The configuration file might not be synchronized with the database.

System action: The keystore is not found.

Administrator response: In the graphical user interface, access the Keystore configuration page. Determine whether the value on the page is the same as the value of the keystore name specified for the config.keystore.name property in the SKLMConfig.properties file. If the values are different, manually change the value of the property to match the value of the name in the graphical user interface, and restart the IBM Security Key Lifecycle Manager server. Then, try the operation again.

CTGKS0001W Could not determine key encoding to obtain key size to validate.

Explanation: Some of the keys in a default key group cannot be validated. These cannot be used for LTO drives. However, the key group may still be valid.

System action: Ensure that the default key group is valid and that LTO drives can be supported.

Administrator response: Ensure that the default key group is valid and that LTO drives can be supported.

CTGKS0002E TCP Listener failed to come up.

Explanation: Only one process can use the TCP port that the TCP Transport Listener requires to run. Another process has the port. Alternatively, the socket timed out.

System action: The TCP Transport Listener is not available to the IBM Security Key Lifecycle Manager server.

Administrator response: Ensure that no other program is using the TCP port. If the other process must have the port, specify a new TCP port number. Then, restart the IBM Security Key Lifecycle Manager server. If the problem continues, examine the audit log. After making corrections, restart the IBM Security Key Lifecycle Manager server. You might need to contact IBM Support.

CTGKS0003E SSL Listener failed to come up.

Explanation: Only one process can use the SSL port that the SSL Transport Listener requires to run. Another process has the port.

System action: The SSL Transport Listener is not available to the IBM Security Key Lifecycle Manager server.

Administrator response: Ensure that no other program is using the SSL port. If the other process must have the port, specify a new Transport Listener SSL port number. Then, restart the IBM Security Key Lifecycle Manager server. If the problem continues, you might need to contact IBM Support.

CTGKS0004E Cannot obtain an instance of SecurityEventHandler.

Explanation: In the SKLMConfig.properties file, the value for the Audit.handler.class property for a distributed system must be com.ibm.tklm.common.audit.file.SimpleFileSecurityEventHandler.

System action: The operation fails.

Administrator response: Determine whether the Audit.handler.class property specifies the correct default value, which should not be changed from the default value. Then, try the operation again.

CTGKS0005E SSL keymanagers failed to load.

Explanation: Ensure that the specified SSL certificate actually exists in the keystore. Alternatively, the SSL certificate might be in an expired state.

System action: The operation fails.

Administrator response: Specify a valid SSL certificate that is not expired. Then, try the operation again.

CTGKS0006E Problem starting key server.

Explanation: The key server internal component that is provided by the IBM Security Key Lifecycle Manager server did not start.

System action: The internal component does not start.

Administrator response: First, examine the audit log for exception information about the key server. You might need to contact IBM Support.

CTGKS0007E Error in property value for:

Explanation: The value that is currently specified is not valid for the property.

System action: The operation fails.

Administrator response: Refer to documentation for the property. The value of the property is specified in the SKLMConfig.properties file. Specify a different value. Then, try the operation again. Changing some properties requires that you restart the IBM Security Key Lifecycle Manager server.

CTGKS0008E debug not initialized.

Explanation: The files that are specified for debug output might be specified as read-only.

System action: Initialization fails.

Administrator response: Specify that the debug output files are writable. Then, try the operation again.

CTGKS0009E TKLMKeyManager not initialized.

Explanation: The class TKLMKeyManager failed to initialize. The SSL port is not functional.

System action: Keys and certificates are not available for key serving.

Administrator response: Ensure that the SSL certificate is correctly configured in the config.keystore.ssl.certalias property. You might also examine the audit log for more information. Make any necessary corrections. Then, try the operation again.

CTGKS0010E TCP Listener went down.

Explanation: Only one process can use the TCP port that the TCP Transport Listener requires to run. Another process has the port. Alternatively, the socket timed out.

System action: The TCP Transport Listener is not available to the IBM Security Key Lifecycle Manager server.

Administrator response: Ensure that no other program is using the TCP port. If the other process must have the port, specify a new TCP port number. Then, restart the IBM Security Key Lifecycle Manager server. If the problem continues, examine the audit log. If no audit exception information is helpful, you might need to contact IBM Support.

CTGKS0011E SSL Listener went down.

Explanation: The SSL Listener that IBM Security Key Lifecycle Manager server provides has failed, possibly because the SSL socket timed out, or because a port conflict occurred.

System action: The operation fails.

Administrator response: To restart the SSL Listener, you must restart the IBM Security Key Lifecycle Manager server. You might need to change the value of the port or the timeout interval. Then, try the operation again. If the SSL socket continues to time out again, or a port conflict continues, you might need to contact IBM Support.

CTGKS0012E The keystore name is null.

Explanation: The key server component cannot locate the IBM Security Key Lifecycle Manager keystore.

System action: IBM Security Key Lifecycle Manager cannot serve the keys.

Administrator response: Ensure that the IBM Security Key Lifecycle Manager keystore is specified, and that the keystore exists. You might run the tklmKeyStoreList command to list the IBM Security Key Lifecycle Manager keystore, or examine the value of the config.keystore.name property in the SKLMConfig.properties file. If necessary, use the tklmKeyStoreAdd command to add a keystore. Then, try the operation again.

CTGKS0013E The keystore unique identifier is null.

Explanation: No value was provided for the keystore Universal Unique Identifier (storeUuid).

System action: IBM Security Key Lifecycle Manager cannot serve the keys.

Administrator response: If you are running the tklmKeyStoreList command, you might alternatively specify the value of the keystore name. If you do not specify a value for either a keystore name or the Universal Unique Identifier, the command lists all keystores.

CTGKS0014E The drive serial number is null.

Explanation: No value was provided for the device serial number.

System action: No device is found.

Administrator response: Specify a value for a valid device serial number that is 12 characters in length. Then, try the operation again.

CTGKS0015E The device with the device serial number *VALUE_0* **does not exist in the database.**

Explanation: An incorrect value was provided for a device serial number.

System action: The create operation fails.

Administrator response: Ensure that you specified the correct device serial number. You might use the tklmDeviceList command to list the devices in the IBM Security Key Lifecycle Manager database. Then, try the operation again.

CTGKS0016E No attributes were specified for the device update operation.

Explanation: No attribute-value pairs were specified to update information for a device.

System action: IBM Security Key Lifecycle Manager cannot serve the keys.

Administrator response: Collect available audit log information and contact IBM Support.

CTGKS0017E A certificate encoding exception occurred when converting the certificate to binary form. The device metadata could not be created for the device with the device serial number *VALUE_0* **.**

Explanation: An internal error occurred.

System action: The create device metadata operation fails.

Administrator response: Collect available audit log information and contact IBM Support.

CTGKS0018E A certificate encoding exception occurred when converting the certificate to binary form. The device metadata could not be updated for the device with the device serial number *VALUE_0* **.**

Explanation: An internal error occurred.

System action: The update device metadata operation fails.

Administrator response: Collect available audit log information and contact IBM Support.

CTGKS0019E The device with the device serial number *VALUE_0* does not exist in the database.

Explanation: An incorrect value was provided for a device serial number.

System action: The delete operation fails.

Administrator response: Ensure that you specified the correct device serial number. You might use the tklmDeviceList command to list the devices in the IBM Security Key Lifecycle Manager database. Then, try the operation again.

CTGKS0021E The key server has no keystore defined. Keys cannot be served to devices.

Explanation: The key server component cannot locate the IBM Security Key Lifecycle Manager keystore.

System action: Keys cannot be served to devices.

Administrator response: Ensure that the IBM Security Key Lifecycle Manager keystore is specified, and that the keystore exists. You might run the tklmKeyStoreList command to list the IBM Security Key Lifecycle Manager keystore, or examine the value of the config.keystore.name property in the SKLMConfig.properties file. If necessary, use the tklmKeyStoreAdd command to add a keystore. Then, try the operation again.

CTGKS0022E KeyGroup specified in symmetricKeySet alias is not valid. Either this key group does not exist or it does not have valid active symmetric keys.

Explanation: The key group does not exist or it does not have valid, active symmetric keys.

System action: No keys are served from the key group.

Administrator response: Ensure that the value is valid for the key group specified by the symmetricKeySet property in the SKLMConfig.properties file. Additionally, ensure that the keys are valid and active, and that they are in the keystore that is specified by the config.keystore.name property. Then, try the operation again.

CTGKS0023E Keys will not be served to LTO devices.

Explanation: No value is set for the symmetricKeyset property in the SKLMConfig.properties file.

System action: No keys are served to LTO Ultrium 4 tape drives.

Administrator response: Specify a valid value for the symmetricKeySet property in the SKLMConfig.properties file. Then, try the operation again.

CTGKS0024E symmetricKeyset must contain valid string with valid key alias. Valid symmetric key aliases are <= 12 characters or exactly 21 characters.

Explanation: Valid symmetric key aliases are less than or equal to 12 characters, or exactly 21 characters.

System action: No keys are served to LTO Ultrium 4 tape drives.

Administrator response: Specify valid values for the keyAliasList parameter of the symmetricKeyset property in the SKLMConfig.properties file. If this is a manual change, you must restart the IBM Security Key Lifecycle Manager server. Then, try the operation again.

CTGKS0025E Error in symmetricKeySet alias range: Make sure the second number in the range is larger than the first.

Explanation: The second number in the alias range must be larger than the first number.

System action: No keys are served to LTO tape drives.

Administrator response: Specify a valid range for the keyAliasList parameter of the symmetricKeyset property in the SKLMConfig.properties file. If this is a manual change, you must restart the IBM Security Key Lifecycle Manager server. Then, try the operation again.

CTGKS0026E Error in symmetricKeySet aliases or key algorithm.

Explanation: The symmetricKeySet property does not have a valid key specification.

System action: No keys are served.

Administrator response: Ensure that the symmetricKeySet parameter points to valid keys. If the zOSCompatibility flag is set on, then the valid algorithm for symmetric keys is DESede. Otherwise the valid algorithm is AES.

CTGKS0027E No valid DKI Aliases specified.Add AES or DESede symmetric keys to symmetricKeySet to support LTO drives.

Explanation: A data key identifier alias is used only for an LTO tape drive. No valid keys of the necessary type are specified in the symmetricKeyset property in the SKLMConfig.properties file.

System action: No keys are served to LTO tape drives.

Administrator response: Add a valid range of AES or DESede symmetric keys to the symmetricKeySet property. Restart the IBM Security Key Lifecycle

Manager server. Then, try the operation again.

CTGKS0028E Alias *VALUE_0* was not found in the keystore.

Explanation: The alias pointed to by the symmetricKeySet property does not exist in the keystore.

System action: The key is not served.

Administrator response: Ensure that a valid key alias is specified. You might examine the current alias values that are specified for the symmetricKeySet property. Then, try the operation again.

CTGKS0029E Alias *VALUE_0* will not be served to LTO drives.

Explanation: The alias pointed to by the symmetricKeySet property does not exist in the keystore.

System action: The key is not served.

Administrator response: Ensure that a valid key alias is specified. You might examine the current alias values that are specified for the symmetricKeySet property. Then, try the operation again.

CTGKS0030E Alias *VALUE_0* was in the keystore but is not a Symmetric KeyEntry.

Explanation: The key alias pointed to by the symmetricKeySet property was found in the keystore, but the key is not a symmetric key.

System action: The key is not served.

Administrator response: Ensure that the key is a symmetric key and that the key is in active state. Then, try the operation again.

CTGKS0031E PKCS11Impl keystore type is not supported.

Explanation: The symmetricKeySet property is pointing to PKCS11Impl keys, which are not supported. The supported algorithms are AES or DESede.

System action: The key operation fails.

Administrator response: Ensure that you are using symmetric keys that conform to the AES algorithm with a size of 256 bits or DESede algorithm with a size of 163 bits. Then, try the operation again.

CTGKS0032E Could not determine key encoding to obtain key size.

Explanation: This is an internal error. Processing could not determine the key encoding.

System action: The key operation fails.

Administrator response: Ensure that you are using symmetric keys that conform to the AES algorithm with a size of 256 bits or DESede algorithm with a size of 163 bits. Then, try the operation again.

CTGKS0033E Expected AES key size is 32 bytes.

Explanation: IBM Security Key Lifecycle Manager supports Advanced Encryption Standard (AES) key that are 32 bytes in length. This key has a different length.

System action: The key operation fails.

Administrator response: Ensure that you are using symmetric keys that conform to the AES algorithm with a size of 32 bytes. Then, try the operation again.

CTGKS0034E Cannot find Secretkey in the keystore with key alias *VALUE_0*

Explanation: Processing cannot find the symmetric key in the keystore with the specified alias.

System action: The key is not served.

Administrator response: Ensure that a valid key alias is specified. You might examine the current alias values that are specified for the symmetricKeySet property. Then, try the operation again.

CTGKS0035E Unsupported algorithm *VALUE_0* .

Explanation: PKCS11Impl is not a supported algorithm for symmetric keys. A supported algortihm is AES.

System action: Make sure to have symmetric keys with AES algorithm and size 32 bytes to support LTO drives.

Administrator response: Make sure to have symmetric keys with AES algorithm and size 32 bytes to support LTO drives.

CTGKS0036E AES key size is *VALUE_0* bytes. Only 32 bytes keys are supported.

Explanation: Supported algorithm is AES with 32 byte size.

System action: symmetricKeySet needs to have valid AES keys. Make sure that 32 byte AES symmetric keys exist in the keystore.

Administrator response: Add 32 byte size AES keys using the graphical user interface or command line interface to support LTO drives.

CTGKS0037E Internal Error:Crypto not initialized. ErrorCode=0xEE0F.

Explanation: Internal error. Crypto class is not initialized. key server cannot serve the keys. ErrorCode is 0xEE0F.

System action: Make sure the keystore type is supported by IBM Security Key Lifecycle Manager.

Administrator response: Check logs for more information. Try to restart IBM Security Key Lifecycle Manager and retry the operation.

CTGKS0038E Vendor ID *VALUE_0* error. ErrorCode=0xEE02.

Explanation: This OEM vendor is not supported by IBM.

System action: Contact IBM Support.

Administrator response: Contact IBM Support.

CTGKS0039E Drive with device serial number *VALUE_0* and WWN *VALUE_1* not found.

Explanation: This drive is not found in the database and cannot be served keys.

System action: No keys are served.

Administrator response: Set the device.AutoPendingAutoDiscovery attribute to a value of 1 for the device group. Then, try the operation again.

CTGKS0040E Socket timed out.

Explanation: Socket time out occurred. It may not be an error. If there are no other errors, no action is necessary.

System action: Socket timed out.

Administrator response: If there are no other errors, no action is necessary.

CTGKS0041E Bad ASC and ASCQ received.

Explanation: Message processing failed. The drive sent bad ASC and ASQ codes.

System action: Cannot serve the keys to this device.

Administrator response: Check the logs for more information. Retry the operation with valid values of ASC and ASCQ.

CTGKS0042E Unexpected payload.

Explanation: Message processing failed. The drive sent a message out of order.

System action: Cannot serve the keys to this device.

Administrator response: Check the logs for more information.

CTGKS0043E Drive certificate type not provided.

Explanation: Message processing failed. The drive did not send certificate type and cannot be validated.

System action: Cannot serve the keys to this device.

Administrator response: Check the logs for more information.

CTGKS0044E No drive certificate provided.

Explanation: Message processing failed. The drive did not send any certificate and cannot be validated.

System action: Cannot serve the keys to this device.

Administrator response: Check the logs for more information.

CTGKS0045E Unsupported drive certificate type.

Explanation: Message processing failed. The drive provided a certificate type that is not valid.

System action: Cannot serve the keys to this device.

Administrator response: Check the logs for more information.

CTGKS0046E Drive certificate not signed properly.

Explanation: Message processing failed. The drive certificate needs to be signed by trusted party.

System action: Cannot serve the keys to this device.

Administrator response: Check the logs for more information.

CTGKS0047E Unexpected DSK count: 0

Explanation: Message processing failed. The drive did not send any certificates.

System action: Cannot serve the keys to this device.

Administrator response: Check the logs for more information.

CTGKS0048E No signature in DSK.

Explanation: Message processing failed. Signature is missing in drive certificate.

System action: Cannot serve the keys to this device.

Administrator response: Check the logs for more information.

CTGKS0049E Cannot verify signature on DSK.

Explanation: Message processing failed. Drive certificate signature cannot be verified.

System action: Cannot serve the keys to this device.

Administrator response: Check the logs for more information.

CTGKS0050E No label expected in UKI with ukiType 0x1931 but label was received.

Explanation: Message processing fails.

System action: Cannot serve the keys to this device.

Administrator response: Check the logs for more information.

CTGKS0051E Expected uki 0x1930 or 0x1931. Uki received is *VALUE_0*

Explanation: Message processing fails.

System action: Cannot serve the keys to this device.

Administrator response: Check the logs for more information.

CTGKS0052E Could not choose the key from the specified key group.

Explanation: Message processing failed. Could not find valid key from the specified key group or key group is not specified. chooseDKI() returned null alias

System action: Cannot serve the keys to this device.

Administrator response: Check the logs for more information. Add valid symmetric keys to the default key group and retry the operation.

CTGKS0053E Keystore did not return SecretKey corresponding to DKI *VALUE_0*

Explanation: Internal Error: Message processing failed. Could not find valid symmetric key from the keystore for the specified key.

System action: The specified key cannot be served to this device.

Administrator response: Check the logs for more information. Make sure key pointed to by this DKI exists in the keystore and retry the operation.

CTGKS0054E Cannot retrieve certificate with label *VALUE_0*

Explanation: Message processing failed. Cannot retrieve certificate with this alias from the keystore.

System action: Cannot serve the keys to this device.

Administrator response: Make sure this certificate exists in the keystore. Check logs for more information.

CTGKS0055E Cannot obtain SKI from the certificate.

Explanation: Message processing failed. Cannot obtain SKI from the provided certificate.

System action: Cannot serve the keys to this device.

Administrator response: Check the logs for more information.

CTGKS0056E Certificate with alias *VALUE_0* **does not exist in the keystore or is incorrect.**

Explanation: Message processing failed. Certificate either does not exist or is not valid. That is, the certificate might have been expired or compromised.

System action: Cannot serve the keys to this device.

Administrator response: Use the command line interface to make sure the certificate with this alias is active and not expired or compromised. Check the logs for more information.

CTGKS0057E Certificate alias is null.

Explanation: Message processing failed. Drive did not send certificate alias.

System action: Check the logs for more information.

Administrator response: Use the command line interface to make sure the certificate is active and not expired. Check the logs for more information.

CTGKS0058E No private key found.

Explanation: Message processing failed. No private key found to decrypt the DK.

System action: The key cannot served to this device.

Administrator response: Check the logs for more information. Make sure keystore contains keys for provided aliases and retry the operation.

CTGKS0059E Unknown payload type received.

Explanation: Message processing failed. Unknown payload type received, the message cannot be parsed.

System action: Cannot serve the keys to the device.

Administrator response: Use the command line interface to make sure the certificate is active and not expired. Check the logs for more information.

CTGKS0060E DKI *VALUE_0* **not found in the keystore.**

Explanation: Message processing failed. DKI not found in the keystore.

System action: Cannot serve the leys to the device.

Administrator response: Make sure the symmetric key with this alias exists in the keystore. Check the logs for more information.

CTGKS0061E eedkuki is null

Explanation: Message processing failed. eedkuki is null.

System action: Cannot serve the leys to the device.

Administrator response: Check the logs for more information.

CTGKS0062E Cannot retrieve key data.

Explanation: Message processing failed. DK cannot be decrypted.

System action: Cannot serve the keys to the device.

Administrator response: Check the logs for more information.

CTGKS0063E Action value *VALUE_0* is incorrect.

Explanation: Message processing failed. Action value incorrect, error in parsing the message.

System action: Cannot serve the keys to the device.

Administrator response: Check the logs for more information.

CTGKS0064E IBM Security Key Lifecycle Manager Truststore does not exist.

Explanation: The truststore file tklmTruststore.jceks does not exist.

System action: The software cannot validate device certificates or serve keys.

Administrator response: If IBM Security Key Lifecycle Manager installed successfully, the tklmTruststore.jceks file should exist. If the file does not exist, contact IBM Support.

CTGKS0065E IBM Security Key Lifecycle Manager Truststore cannot be loaded.

Explanation: The IBM Security Key Lifecycle Manager Truststore file tklmTruststore.jceks cannot be loaded.

System action: Device certificates cannot be validated. Keys cannot be served.

Administrator response: Make sure that the file exists and that the password of the file has not been changed. Check the logs for more information. If that does not help, call IBM Support.

CTGKS0066E symmetricKeySet is incorrect.

Explanation: Configuration property symmetricKeySet cannot be validated. LTO drives cannot be served.

System action: If symmetrickeySet points to a key group, then make sure the group has at least one AES or DESede key depending on whether the zOSCompatibility flag is off or on. If symmetricKeySet points to a key alias, then make sure the alias exists in the keystore and is valid.

Administrator response: Refer to the product documentation on how to create key groups and symmetric keys.

CTGKS0067E No keys available in key group *VALUE_0*.

Explanation: All the keys from this key group are already served to drives and no more unique keys are available to serve to the LTO drive.

System action: With stopRoundRobinKeyGrps flag on, the keys from the key group can be used only once. There are no more keys available to serve to the LTO drive. Drive write fails with error 0XEE34.

Administrator response: Refer to the product documentation on how to create key groups and symmetric keys. Add more keys to this key group and retry the operation.

CTGKS0068E Server parameters not initialized.

Explanation: key server is not able to read the keystore password from the database and server parameters cannot be initialized.

System action: IBM Security Key Lifecycle Manager will not be able to serve keys to devices.

Administrator response: Make sure keystore is properly configured in the IBM Security Key Lifecycle Manager server and the internal property tklm.encryption.password is present in the SKLMConfig.properties file. Refer to the logs for more information. Correct the problem and restart the server.

CTGKS0069E Client certificate chain not received.

Explanation: SSL connection fails because the server did not receive any certificate from a client to be able to authenticate that client. This error can happen only if clientAuthentication is set to 2 (required) in SKLMConfig.properties file for key server. Note that for KMIP protocol, clientAuthentication is always set to required.

System action: SSL handshake fails and SSL connection cannot be established.

Administrator response: Make sure client is configured to send a certificate to IBM Security Key

Lifecycle Manager that IBM Security Key Lifecycle Manager can trust. These trusted SSLClient certificates can be listed with the tklmCertList command. Refer to the logs for more information. Correct the problem and restart the server.

CTGKS0070E Server does not trust the client certificate.

Explanation: Client authentication fails because the server does not trust the certificate sent by the client. This error can happen only if clientAuthentication is set to 2 (required) in the SKLMConfig.properties file for key server. Note that for KMIP protocol, clientAuthentication is always set to required.

System action: SSL handshake fails and SSL connection cannot be established.

Administrator response: Make sure client is configured to send a certificate to IBM Security Key Lifecycle Manager that IBM Security Key Lifecycle Manager can trust. These trusted SSLClient certificates can be listed with the tklmCertList command. Refer to the logs for more information. Correct the problem and restart the server.

CTGKS0071E No SSLServer certificate with alias *VALUE_0* found in the database.

Explanation: The SSL certificate specified by config.keystore.ssl.certalias in the SKLMConfig.properties file is not found in the database or is not marked as the SSLServer certificate.

System action: SSL handshake fails and SSL connection cannot be established.

Administrator response: Make sure the SSL server certificate is configured and exists in the IBM Security Key Lifecycle Manager database by listing it with tklmCertList command. Once the correct SSL server is configured then restart the server. Refer to the logs for more information.

CTGKS0072E Certificate alias length cannot exceed 256 characters.

Explanation: Certificate alias exceeded 256 characters in length.

System action: Certificate creation fails.

Administrator response: Check the logs for more information.

CTGKS0073E Attribute *VALUE_0* is not supported for device group *VALUE_1* .

Explanation: Attribute is not supported for the specified device group.

System action: The operation fails.

Administrator response: Check the logs for more information.

CTGKS0074E Message signature does not verify.

Explanation: Message signature cannot be verified. Message processing failed. key server cannot serve the keys.

System action: Cannot serve the keys to this device.

Administrator response: Check the logs for more information. Try to restart IBM Security Key Lifecycle Manager and retry the operation.

CTGKS0075E Message type not *VALUE_0*.

Explanation: Received wrong message type for the message, message cannot be parsed. Message processing failed. key server cannot serve the keys.

System action: Cannot serve the keys to this device.

Administrator response: Check the logs for more information. Try to restart IBM Security Key Lifecycle Manager and retry the operation.

CTGKS0076E DKI length not 12 bytes or data key length not 32 bytes in KADDescriptor.

Explanation: Message processing failed. key server cannot serve the keys.

System action: Cannot serve the keys to this device.

Administrator response: Check the logs for more information. Try to restart IBM Security Key Lifecycle Manager and retry the operation.

CTGKS0077E Audit or Configuration objects are null.

Explanation: Internal error. Audit or Configuration objects are not initialized for the object being used.

System action: The system cannot process the message.

Administrator response: Refer to the logs for more information.

CTGKS0078E Keystore not found in the database.

Explanation: The keystore is not found in the database.

System action: The system cannot initialize and function properly.

Administrator response: Make sure the database server is up and running. Refer to the logs for more information.

CTGKS0079E Unknown message type *VALUE_0*.

Explanation: Message processing failed. Unknown message type received, cannot parse the message.

System action: Cannot serve the keys to this device.

Administrator response: Check the logs for more information. Retry the operation with valid values of message type.

CTGKS0080E Cannot update device labels or type for this device *VALUE_0*.

Explanation: Message processing failed. Internal error occurred while updating device metadata.

System action: The device metadata will not be updated in the database.

Administrator response: Check the logs for more information. Retry the operation with valid values of message type.

CTGKS0081E audit not initilized.

Explanation: The files that are specified for audit output might be specified as read-only.

System action: Initialization fails.

Administrator response: Specify that the audit output files are writable. Then, try the operation again.

CTGKS0124E Migration fails. The file or directory {0} does not exist.

Explanation: The migration program accesses files from the previous and new IBM Security Key Lifecycle Manager directory. One of the critical files or the directory cannot be accessed.

System action: The migration program fails.

Administrator response: Verify that the file or directory exists and has the appropriate read and write permissions. Run the migration program again.

CTGKS0125E Migration fails. The specified argument {0} is not a directory.

Explanation: The argument specified must be a directory where Tivoli Integrated Portal is installed.

System action: The migration program fails.

Administrator response: Specify a valid directory where Tivoli Integrated Portal is installed. Run the migration again.

CTGKS0126E Migration fails. The file {0} could not be read. The exception {1} occurred.

Explanation: Before starting the migration process, the migration program reads the configuration file to verify that all the critical information needed to migrate the previous IBM Security Key Lifecycle Manager is available and correct. This file could not be read.

System action: The migration program fails.

Administrator response: Verify that the file exists and has correct read permissions. This file might have been removed. If the file does not exist, restore the backed up version of IBM Security Key Lifecycle Manager. Run the migration program again.

CTGKS0127E Migration fails. The property file {0} is missing the required property {1}.

Explanation: Before starting the migration process, the migration program reads the configuration file to verify that all the critical information needed to migrate the previous IBM Security Key Lifecycle Manager is available and correct. One of the required properties is missing.

System action: The migration program fails.

Administrator response: The required property might have been removed. add the property with the correct value or restore the backed up version of IBM Security Key Lifecycle Manager. Run the migration program again.

CTGKS0128E Migration fails. The value {0} is not a valid value for the configuration parameter {1}.

Explanation: Before starting the migration process, the migration program reads the configuration file to verify that all the critical information needed to migrate the previous IBM Security Key Lifecycle Manager is available and correct. One of the required properties has an incorrect value.

System action: The migration program fails.

Administrator response: The required property might have been modified. Either set the property to the correct value or restore the backed up version of IBM Security Key Lifecycle Manager. Run the migration program again.

CTGKS0129E Migration fails. IBM Security Key Lifecycle Manager could not validate the previous Tivoli Integrated Portal Administrator password. The password might be incorrect or the server might not be running.

Explanation: Before starting the migration process, the migration program validates the Tivoli Integrated Portal Administrator password. The Tivoli Integrated

Portal password is required to start and stop the Tivoli Integrated Portal server and to undeploy the previous IBM Security Key Lifecycle Manager.

System action: The migration program fails.

Administrator response: Verify that Tivoli Integrated Portal Server is running and the password is correct. Place quote marks around the password if necessary. Run the migration program again.

CTGKS0130E Migration fails. IBM Security Key Lifecycle Manager could not validate the database administrator password. The password might be incorrect or the database server might not be running.

Explanation: Before starting the migration process, the migration program validates the database administrator password. The password is required to migrate the database schema and the data.

System action: The migration program fails.

Administrator response: Verify that the database server is running and the password specified is correct. Run the migration program again.

CTGKS0131E Migration fails. The database schema could not be migrated to the latest version of IBM Security Key Lifecycle Manager.

Explanation: Before starting the migration process, the migration program validates that the database schema is at the earlier level of IBM Security Key Lifecycle Manager.

System action: The migration program fails.

Administrator response: Verify that the database server is running and the correct version of IBM Security Key Lifecycle Manager is installed. Run the migration program again.

CTGKS0132E Migration fails. IBM Security Key Lifecycle Manager could not validate the new Tivoli Integrated Portal Administrator password. The password might be incorrect or the server might not be running.

Explanation: Before starting the migration process, the migration program validates the Tivoli Integrated Portal Administrator password. The Tivoli Integrated Portal password is required to start and stop the new Tivoli Integrated Portal server.

System action: The migration program fails.

Administrator response: Verify that the new Tivoli Integrated Portal Server is running and the specified password is correct. Place quote marks around the password if necessary. Run the migration program again.

CTGKS0133E Migration fails. IBM Security Key Lifecycle Manager could not validate the new IBM Security Key Lifecycle Manager Administrator password. The password might be incorrect or the server might not be running.

Explanation: Before starting the migration process, the migration program validates the Tivoli Integrated Portal Administrator password. The Tivoli Integrated Portal password is required to migrate the previously scheduled rollover tasks.

System action: The migration program fails.

Administrator response: Verify that the new Tivoli Integrated Portal Server is running and the specified password is correct. Place quote marks around the password if necessary. Run the migration program again.

CTGKS0134E Migration fails. The IBM Security Key Lifecycle Manager database password contains characters that are other than [a-z,A-Z,0-9].

Explanation: Before starting the migration process, the migration program validates that the database administrator password contains only allowable characters.

System action: The migration program fails.

Administrator response: Verify that the database password contains only characters [a-z,A-Z,0-9]. Change the password to include only characters[a-z,A-Z,0-9] using tools that the operating system provides and run the migration again. After successful migration, you might change the database password as documented in the IBM Security Key Lifecycle Manager Installation and Configuration Guide.

CTGKS0135E Migration fails. IBM Security Key Lifecycle Manager cannot verify that schema is at the appropriate level before migration starts. The database server might not be running.

Explanation: Before starting the migration process, the migration program validates that the database schema is at the correct level.

System action: The migration program fails.

Administrator response: Verify that the database server is running. Run the migration program again.

CTGKS0136E Migration fails. IBM Security Key Lifecycle Manager cannot migrate the key groups. The database server might not be running, the transaction log might be full, or an unexpected database error occurred.

Explanation: The migration program migrates the existing groups to assign device groups to each of the key groups.

System action: The migration program fails.

Administrator response: Verify that the database server is running and transaction log is not full. The database exception might provide additional information about the problem. Correct the condition and run the migration program again. If the problem persists, contact IBM Support.

CTGKS0137E Migration fails. IBM Security Key Lifecycle Manager cannot migrate the keys. The database server might not be running, the transaction log might be full or an unexpected database error occurred.

Explanation: The migration program migrates the existing keys to assign device groups to each of the keys and other data transformations needed to make the keys work with the latest version of IBM Security Key Lifecycle Manager.

System action: The migration program fails.

Administrator response: Verify that the database server is running and transaction log is not full. The database exception may provide additional information about the problem. Correct the condition and run the migration program again. If the problem persists, contact IBM Support.

CTGKS0138E Migration fails. IBM Security Key Lifecycle Manager cannot migrate the certificates. The database server might not be running, the transaction log might be full, or an unexpected database error occurred.

Explanation: The migration program migrates the existing certificates to assign device groups to each of the certificates and other data transformations needed to make the certificates work with the latest version of IBM Security Key Lifecycle Manager.

System action: The migration program fails.

Administrator response: Verify that the database server is running and transaction log is not full. The database exception may provide additional information about the problem. Correct the condition and run the migration program again. If the problem persists, contact IBM Support.

CTGKS0139E Migration fails. IBM Security Key Lifecycle Manager cannot migrate the device attributes. The database server might not be running or an unexpected database error occurred.

Explanation: The migration program sets the device attributes on how to handle unknown DS8000 and LTO drives based on the properties in the SKLMConfig.properties file.

System action: The migration program fails.

Administrator response: Verify that the database server is running. The database exception may provide additional information about the problem. Correct the condition that caused the error and run the migration program again. If the problem persists, contact IBM Support.

CTGKS0140E Migration fails. IBM Security Key Lifecycle Manager cannot migrate the device audit data. The server might not be running, the transaction log might be full, or an unexpected database error occurred.

Explanation: The migration program migrates the existing device audit metadata to assign device groups to each of the audit records.

System action: The migration program fails.

Administrator response: Verify that the database server is running and the transaction log is not full. The database exception may provide additional information about the problem. Correct the condition and run the migration program again. If the problem persists, contact IBM Support.

CTGKS0141E Migration fails. IBM Security Key Lifecycle Manager cannot migrate the scheduled tasks. The exception {0} occurred. The database server might not be running, an unexpected database error occurred, or the Tivoli Integrated Portal server might not be running.

Explanation: The migration program migrates the existing scheduled rollover tasks to assign device groups to each of the tasks.

System action: The migration program fails.

Administrator response: Verify that the database server and the Tivoli Integrated Portal server are running. The database exception may provide additional information about the problem. Correct the condition and run the migration program again. If the problem persists, contact IBM Support.

CTGKS0142E Migration fails. You must be running Encryption Key Manager Version 2.1 before migrating to IBM Security Key Lifecycle Manager.

Explanation: The migration program verifies that Encryption Key Manager Version 2.1 is installed. The properties TransportListener.ssl.keystore.password.obfuscated and config.keystore.password.obfuscated must be set in the KeyManagerConfig.properties file.

System action: The migration program fails.

Administrator response: Verify that Encryption Key Manager Version 2.1 is installed. If you are running an earlier version of Encryption Key Manager, upgrade to Version 2.1 and run the migration again.

CTGKS0143E Server parameters not initialized.

Explanation: The key group cannot be deleted because server parameters are not initialized.

System action: The delete key group operation fails.

Administrator response: The key group cannot be deleted because server parameters are not initialized.

CTGKS0146E The message type is not as expected: *VALUE_0* **.**

Explanation: The message type is not as expected.

System action: The operation fails

Administrator response: Examine the exception message, and then try the operation again.

CTGKS0147E Message OEM shared secret does not verify.

Explanation: Message OEM shared secret does not verify.

System action: The operation fails.

Administrator response: Make sure the OEM shared secret is correct, and try again.

CTGKS0148W Warning: The migration to IBM Security Key Lifecycle Manager was successful. However, the migration could not remove the {0} component of IBM Security Key Lifecycle Manager Version 1 from Tivoli Integrated Portal Server. Refer to the IBM Security Key Lifecycle Manager Installation and Configuration Guide for steps to manually remove remaining components of IBM Security Key Lifecycle Manager Version 1 from Tivoli Integrated Portal Server.

Explanation: The migration program, after successfully migrating to IBM Security Key Lifecycle Manager Version 2, removes IBM Security Key Lifecycle Manager Version 1 from Tivoli Integrated Portal server. The components are the graphical user interface, data sources, and application. The process could not remove all the components of IBM Security Key Lifecycle Manager Version 1.

System action: The migration program succeeds with a warning.

Administrator response: Verify that the IBM Security Key Lifecycle Manager Version 2 is functioning correctly. Refer to the IBM Security Key Lifecycle Manager Version 2 Installation and Configuration Guide for steps to remove the remaining components of IBM Security Key Lifecycle Manager Version 1 from Tivoli Integrated Portal Server.

CTGKS0149E Migration fails. IBM Security Key Lifecycle Manager could not validate the previous Tivoli Integrated Portal Administrator password. Possible reasons: specified password is incorrect or the server is not running.

Explanation: Before starting the migration process, the migration program validates the Tivoli Integrated Portal Administrator password. The Tivoli Integrated Portal password is required to start and stop the Tivoli Integrated Portal server and to undeploy the previous IBM Security Key Lifecycle Manager.

System action: The migration program fails.

Administrator response: Verify that Tivoli Integrated Portal Server is running and the password specified is correct. Run the migration program again.

CTGKS0150E Usage: migratetklm db_administrator_pwd v1_tipadmin_pwd v2_tipadmin_pwd v2_tklmadmin_pwd\n where \n db_administrator_pwd - IBM Security Key Lifecycle Manager Version 1 database administrator password.\n v1_tipadmin_pwd - Tivoli Integrated Portal Server Administrator password for IBM Security Key Lifecycle Manager Version 1\n v2_tipadmin_pwd - Tivoli Integrated Portal Server Administrator password for IBM Security Key Lifecycle Manager Version 2\n v2_tklmadmin_pwd - IBM Security Key Lifecycle Manager Version 2, Administrator password.

Explanation: The migration program was started with an incorrect number of arguments. Either specify all the required passwords as arguments or start the program without any arguments. You will be prompted for the passwords when the program starts.

System action: The migration program fails.

Administrator response: Start the migration program either specifying all the passwords or without any passwords.

CTGKS0155E Migration fails. The following exception occurred: {0}

Explanation: While performing a migration step, an exception occurred. The migration program immediately prints an error message indicating the correct action.

System action: The migration program fails.

Administrator response: Perform the action that the message provides.

CTGKS0158W Partially removed IBM Security Key Lifecycle Manager Version 1 files. However, you might remove the rest of the files manually.

Explanation: The migration could not remove all files in IBM Security Key Lifecycle Manager Version 1.

System action: Migration succeeds.

Administrator response: Remove the remainder of the files from IBM Security Key Lifecycle Manager Version 1 after moving audit logs to another location.

CTGKS0160E Migration fails. Refer to the SKLM_HOME/migration/migrate.log file to identify causes of failure and recovery steps.

Explanation: Migration fails.

System action: Migration fails.

Administrator response: Perform the recovery step identified by one or more earlier error messages in the migrate.log file.

CTGKS0163E Migration fails. The database schema version could not be set to 2.0 to indicate successful database schema and data migration during the previous migration attempts.

Explanation: At the end of migration, the migration program sets the version to 2.0 in the database to indicate that the schema has been upgraded to the latest level. It also drops the migration-related table from the database.

System action: Migration fails.

Administrator response: The database server might have stopped. Start the database server and run the migration again.

CTGKS0164E Migration fails. The database instance or database could not be migrated to the latest version of DB2.

Explanation: The migration program migrates the IBM Security Key Lifecycle Manager database instance and database to the latest version of DB2. This process failed.

System action: Migration fails.

Administrator response: There might be low disk space or an unexpected error. Run the migration program again. If the problem persists, contact IBM Support.

CTGKS0165E Migration fails. The migration program cannot copy either the user keystore or the IBM Security Key Lifecycle Manager internal keystore to the new location under the IBM Security Key Lifecycle Manager Version 2 folder.

Explanation: The migration program copies the user and internal keystores to the IBM Security Key Lifecycle Manager Version 2 location.

System action: Migration fails.

Administrator response: There might be incorrect permissions for the keystore files in IBM Security Key Lifecycle Manager Version 1 or the target folder does not have correct permissions. Verify that both the source and target directories have correct permissions for copying. Run the migration program again. If the problem persists, contact IBM Support.

CTGKS0168E Migration fails. The user key store {0} cannot be not migrated successfully.

Explanation: The migration program copies the user keystore from IBM Security Key Lifecycle Manager Version 1 to Version 2. This operation did not succeed.

System action: Migration fails.

Administrator response: Verify that permissions for the user keystore have the correct read permissions in IBM Security Key Lifecycle Manager Version 1 and the directory in Version 2 where user keystore would have been copied has the correct write permissions.

CTGKS0171E Migration fails. IBM Security Key Lifecycle Manager cannot migrate the devices.

Explanation: The migration program migrates the existing devices to assign the original device type to each of the devices to be consistent with the latest version of IBM Security Key Lifecycle Manager.

System action: The migration program fails.

Administrator response: Verify that the database server is running and transaction log is not full. The database exception may provide additional information about the problem. Correct the condition and run the migration program again. If the problem persists, contact IBM Support.

CTGKS0173E Migration fails. The migration program cannot migrate the rollover task because of an internal error. Contact IBM Support.

Explanation: The migration program fails.

System action: The migration program fails.

Administrator response: Send the migrate.log to IBM Support.

CTGKS0180E Migration fails. IBM Security Key Lifecycle Manager cannot add missing public keys. The database server might not be running, the transaction log might be full, or an unexpected database error occurred.

Explanation: The migration program fails to add missing public keys to those public/private keypairs that were imported.

System action: The migration program fails.

Administrator response: Verify that the database server is running and transaction log is not full. The database exception might provide additional information about the problem. Correct the condition and run the migration program again. If the problem persists, contact IBM Support.

CTGKS0183E Migration fails. The migration program fails to create a unique database entry for each alias when a key or certificate has multiple aliases. The database server might not be running, the transaction log might be full, or an unexpected database error occurred.

Explanation: The migration program fails to create a unique database entry for each alias when a key or certificate has multiple aliases.

System action: The migration program fails.

Administrator response: Verify that the database server is running and transaction log is not full. The database exception may provide additional information about the problem. Correct the condition and run the migration program again. If the problem persists, contact IBM Support.

CTGKS0191E The migration program failed to execute a batch file or a shell script.

Explanation: The migration program executes one or more shell scripts or batch files during migration. An error occurred.

System action: The migration program fails.

Administrator response: One or more batch files or shell scripts may not have correct permissions. Contact IBM support if problem persists.

CTGKS0500E Unsupported Value received in *VALUE_0* **for** *VALUE_1* **field.**

Explanation: The message from the client has an unsupported value in the given field.

System action: IBM Security Key Lifecycle Manager cannot process the message for the client.

Administrator response: Check the logs for more information.

CTGKS0501E Cannot skip bytes, not enough bytes.

Explanation: The message does not have enough bytes left and cannot skip the bytes to read the whole message.

System action: IBM Security Key Lifecycle Manager cannot process the message for the client.

Administrator response: Check the logs for more information.

CTGKS0503E Integrity check failed.

Explanation: The EncryptedPayload cannot be verified because it failed the integrity check.

System action: IBM Security Key Lifecycle Manager cannot process the message for the client.

Administrator response: Check the logs for more information.

CTGKS0504E No value provided in *VALUE_0* **for** *VALUE_1* **field.**

Explanation: The message from the client did not provide the value in the given field.

System action: IBM Security Key Lifecycle Manager cannot process the message for the client.

Administrator response: Check the logs for more information.

CTGKS0505E Not enough cryptographic algorithm descriptors in UTCryptographicAlgorithmsPayload.

Explanation: The UTCryptographicAlgorithmsPayload does not have enough cryptographic algorithm descriptors. The expected number of descriptors is 2.

System action: IBM Security Key Lifecycle Manager cannot process the message for the client.

Administrator response: Check the logs for more information.

CTGKS0506E Not enough bytes in SPHeader. Number of bytes received are *VALUE_0*, expected number of bytes in SPHeader is 16.

Explanation: Not enough bytes in SPHeader in SPINCommand received.

System action: IBM Security Key Lifecycle Manager cannot process the message for the client.

Administrator response: Check the logs for more information.

CTGKS0507E More than one *VALUE_0* in the *VALUE_1* message.

Explanation: More than one payload received in the message as indicated. Expected to receive only one.

System action: IBM Security Key Lifecycle Manager cannot process the message for the client.

Administrator response: Check the logs for more information.

CTGKS0508E Not enough bytes in SPHeader. Number of bytes received are *VALUE_0*, expected number of bytes in SPHeader is 16.

Explanation: Not enough bytes in SPHeader in SPOUTCommand received.

System action: IBM Security Key Lifecycle Manager cannot process the message for the client.

Administrator response: Check the logs for more information and resend the request.

CTGKS0509E DS_SAI value is larger than 4 bytes. Value received is *VALUE_0*.

Explanation: DS_SAI value cannot exceed 4 bytes.

System action: IBM Security Key Lifecycle Manager cannot process the message for the client.

Administrator response: Check the logs for more information to correct the problem and try sending the request again.

CTGKS0510E Encryption key and/or algorithm is not set.

Explanation: Encryption key and/or algorithm is not set. It needs to be set to AES_CBC. Cannot proceed.

System action: IBM Security Key Lifecycle Manager cannot process the message for the client.

Administrator response: Check the logs for more information to correct the problem and try sending the request again.

CTGKS0511E Ac_SAI field mismatch. Value received from the device is *VALUE_0*.

Explanation: AC_SAI field does not match with the one that is sent to the device.

System action: IBM Security Key Lifecycle Manager cannot process the message for the client.

Administrator response: Check the logs for more information to correct the problem and try sending the request again.

CTGKS0512E UTCryptographicAlgorithmsPayload present in SPINKeyExchangeResponse.

Explanation: UTCryptographicAlgorithmsPayload not expected in SPINKeyExchangeResponse.

System action: IBM Security Key Lifecycle Manager cannot process the message for the client.

Administrator response: Check the logs for more information to correct the problem and try sending the request again.

CTGKS0513E Bad certificate request data.

Explanation: Bad certificate request data received in SPINKeyExchangeResponse.

System action: IBM Security Key Lifecycle Manager cannot process the message for the client.

Administrator response: Check the logs for more information to correct the problem and try sending the request again.

CTGKS0514E IKEV2SERVER certificate not found.

Explanation: IKEV2SERVER certificate not configured on the IBM Security Key Lifecycle Manager server.

System action: IBM Security Key Lifecycle Manager cannot process the message for the client.

Administrator response: Ensure that the IKEV2SEREVR certificate is configured on the server using the graphical user interface or by running the tklmListConfig command. Check that it is marked trusted and not expired. Check the logs for more information to correct the problem and try sending the request again.

CTGKS0515E No certificate request received from drive.

Explanation: Certificate request payload is required in SPINKeyExchangeResponse.

System action: IBM Security Key Lifecycle Manager cannot process the message for the client.

Administrator response: Check the logs for more information to correct the problem and try sending the request again.

CTGKS0516E Illegal certificate encoding *VALUE_0* received in *VALUE_1*.

Explanation: Illegal certificate encoding received. Only X509 encoding is supported.

System action: IBM Security Key Lifecycle Manager cannot process the message for the client.

Administrator response: Check the logs for more information to correct the problem and try sending the request again.

CTGKS0517E IKEV2CLIENT certificate not valid.

Explanation: IKEV2CLIENT certificate configured on the IBM Security Key Lifecycle Manager server but does not seem to be valid.

System action: IBM Security Key Lifecycle Manager cannot process the message for the client.

Administrator response: Ensure that the IKEV2CLIENT certificate is marked trusted and not expired. Check the audit logs for more information to correct the problem and try sending the request again.

CTGKS0518E Identification sent by the device cannot be verified.

Explanation: Idenitification payload send by a client has a server ID that does not match ID in the certificate in SPINAuthenticationResponse. Identification check failed.

System action: IBM Security Key Lifecycle Manager cannot process the message for the client.

Administrator response: Check the logs for more information to correct the problem and try sending the request again.

CTGKS0519E UTCryptographicAlgorithmsPayload in SPINAuthenticationResponse differs from that in SPOUTAuthentication.

Explanation: UTCryptographicAlgorithmsPayload in SPINAuthenticationResponse received from the device has to match the UTCryptographicAlgorithmsPayload sent in SPOUTAuthentication.

System action: IBM Security Key Lifecycle Manager cannot process the message for the client.

Administrator response: Check the logs for more information to correct the problem and try sending the request again.

CTGKS0520E Authentication of authentication payload failed.

Explanation: Signature cannot be verified in SPINAuthenticationResponse. Authentication check failed.

System action: IBM Security Key Lifecycle Manager cannot process the message for the client.

Administrator response: Check the logs for more information to correct the problem and try sending the request again.

CTGKS0522E Nonce minimum length *VALUE_0* is greater than Nonce maximum length allowed which is *VALUE_1*.

Explanation: Nonce minimum length cannot be greater than the maximum length allowed.

System action: IBM Security Key Lifecycle Manager cannot process the message for the client.

Administrator response: Check the logs for more information to correct the problem and try sending the request again.

CTGKS0523E Cannot get public keys for trusted certificates.

Explanation: Cannot get public keys for trusted certificates configured in IBM Security Key Lifecycle Manager. Cannot build SPOUTKeyExchange.

System action: IBM Security Key Lifecycle Manager cannot process the message for the client.

Administrator response: Check the logs for more information to correct the problem and try sending the request again.

CTGKS0524E Wrong message received. At this state *VALUE_0* the message expected is *VALUE_1*.

Explanation: IBM Security Key Lifecycle Manager received a message out of order.

System action: IBM Security Key Lifecycle Manager cannot process the message for the client.

Administrator response: Check the logs for more information to correct the problem and try sending the request again.

CTGKS0525E CCS timeout occurred. The timer started at *VALUE_0* and the current timestamp is *VALUE_1*.

Explanation: IBM Security Key Lifecycle Manager requires the next message to be received within CCS timeout period. The CCS timeout value is configured in SKLMConfig.properties file by config.keystore.IKEV2SERVER.CCSTimeout property name or 5 minutes by default.

System action: IBM Security Key Lifecycle Manager cannot process the message for the client.

Administrator response: Make sure the configured CCS timeout has a suitable value. Check the logs for more information to correct the problem and try sending the request again.

CTGKS0526E Payload not encrypted.

Explanation: Payload in SPINAuthenticationResponse and SPOUTAuthentication are required to be encrypted.

System action: IBM Security Key Lifecycle Manager cannot process the message for the client.

Administrator response: Check the logs for more information to correct the problem and try sending the request again.

CTGKS0527E No private key found to sign the certificate.

Explanation: No private key found to sign the certificate.

System action: IBM Security Key Lifecycle Manager cannot process the message for the client.

Administrator response: Check the logs for more information to correct the problem and try sending the request again.

CTGKS0528E IKEv2Server alias is not defined.

Explanation: IKEv2Server alias needs to be configured.

System action: IBM Security Key Lifecycle Manager cannot process the message for the client.

Administrator response: Configure the IKEv2Server alias using the graphical user interface by going to Advanced Configuration > Server certificate. Check the logs for more information to correct the problem and try sending the request again.

CTGKS0529E Identification payload is not properly encoded.

Explanation: Identification payload is not properly encoded.

System action: IBM Security Key Lifecycle Manager cannot process the message for the client.

Administrator response: Check the logs for more information to correct the problem and try sending the request again.

CTGKS0530E Point coordinates do not match field size.

Explanation: Point coordinates do not match field size. Cannot generate ECPrivateKey.

System action: IBM Security Key Lifecycle Manager cannot process the message for the client.

Administrator response: Check the logs for more information to correct the problem and try sending the request again.

CTGKS0531E Only uncompressed point format supported.

Explanation: Only uncompressed point format supported. Cannot generate ECPrivateKey.

System action: IBM Security Key Lifecycle Manager cannot process the message for the client.

Administrator response: Check the logs for more information to correct the problem and try sending the request again.

CTGKS0532E Point does not match field size.

Explanation: Point does not match field size. Cannot generate ECPrivateKey.

System action: IBM Security Key Lifecycle Manager cannot process the message for the client.

Administrator response: Check the logs for more information to correct the problem and try sending the request again.

CTGKS0533E *VALUE_0* not set.

Explanation: Internal error. The value of the given field is not set.

System action: IBM Security Key Lifecycle Manager cannot process the message for the client.

Administrator response: Check the logs for more information to correct the problem and try sending the request again.

CTGKS0534E One or more key lengths is zero.

Explanation: Internal error. One of the key lengths is not set.

System action: IBM Security Key Lifecycle Manager cannot process the message for the client.

Administrator response: Check the logs for more information to correct the problem and try sending the request again.

CTGKS0535E Private key is larger than D-H modulus.

Explanation: Internal error. Cannot generate D-H private key.

System action: IBM Security Key Lifecycle Manager cannot process the message for the client.

Administrator response: Check the logs for more information to correct the problem and try sending the request again.

CTGKS0536E The alias *VALUE_0* does not identify a certificate in the keystore.

Explanation: Either the alias does not exist in the keystore, or it does not refer to a certificate.

System action: The operation fails.

Administrator response: Specify a valid alias, and try the operation again.

CTGKS0560E Data structure length is incorrect.

Explanation: Data structure length is incorrect.

System action: The operation fails.

Administrator response: Correct the data structure, and try the operation again.

CTGKS0561E Response code is incorrect.

Explanation: Response code is incorrect.

System action: The operation fails.

Administrator response: Try the operation again.

CTGKS0562E Expecting page code *VALUE_0* .

Explanation: Unexpected page code.

System action: The operation fails.

Administrator response: Examine the error message, and then try again.

CTGKS0563E The routing structure type is not as expected: *VALUE_0* .

Explanation: The routing structure type is not as expected.

System action: The operation fails.

Administrator response: Examine the exception message, and then try again.

CTGKS0564E Expecting page code *VALUE_0* .

Explanation: Unexpected page code.

System action: The operation fails.

Administrator response: Examine the error message, and then try again.

CTGKS0565E Unknown signature type.

Explanation: The signature type is unknown.

System action: The operation fails.

Administrator response: Check the signature type, and then try again.

CTGKS0566E Message has been tampered with.

Explanation: Message has been tampered with.

System action: The operation fails.

Administrator response: Make sure the message is correct, and try again.

CTGKS0566E Message has been tampered with.

Explanation: Message has been tampered with.

System action: The operation fails.

Administrator response: Make sure the message is correct, and try again.

KMIP messages

These are the KMIP error messages.

CTGKP0001E Check failed. Nothing to return in response.

Explanation: Internal error. Cannot return response for Check operation.

System action: Cannot process the KMIP message.

Administrator response: Check the audit logs. Correct the problem and retry the operation.

CTGKP0002E Check failed, no unique identifier to return in the response.

Explanation: Internal error. Cannot return response for the Check operation.

System action: Cannot process the KMIP message.

Administrator response: Check the audit logs. Correct the problem and retry the operation.

CTGKP0003E Field *VALUE_0* not specified.

Explanation: Field is not specified in the KMIP message that was received.

System action: Cannot process the message.

Administrator response: Correct the input and retry the operation.

CTGKP0004E Structure *VALUE_0* is empty.

Explanation: Field is expected to have a valid value in the KMIP message that was received.

System action: Cannot process the message.

Administrator response: Correct the input and retry the operation.

CTGKP0005E Structure *VALUE_0* is null.

Explanation: Structure is not specified in the KMIP message that was received.

System action: Cannot process the message.

Administrator response: Correct the input and retry the operation.

CTGKP0006E Header tag is neither Response nor Request. The tag is *VALUE_0*.

Explanation: Incorrect tag received in the KMIP message header.

System action: Cannot process the message.

Administrator response: Correct the input and retry the operation.

CTGKP0007E Protocol Version : major: *VALUE_0* , minor *VALUE_0* is not supported.

Explanation: Incorrect protocol version received in the KMIP message header.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP0008E Message is neither Response nor Request. The message tag is *VALUE_0*.

Explanation: Cannot parse the message. An incorrect KMIP message tag was received.

System action: Cannot process the message.

Administrator response: Correct the input and retry the operation.

CTGKP0009E Message is not a single structure. Type of message object is *VALUE_0*.

Explanation: Cannot parse the message. Received an incorrect KMIP message type.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP0010E Message that was received is null.

Explanation: Nothing to parse. Received no KMIP message.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP0011E Vendor extension tag value is incorrect.

Explanation: Cannot parse the KMIP message. Received an incorrect tag value.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP0012E Operation is pending but no asynchronous correlation value was specified.

Explanation: Cannot proceed. Received no asynchronous correlation value in the KMIP message.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP0014E Following values must all be specified: ***VALUE_0*****.**

Explanation: Cannot proceed. Required values are missing in the message.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP0015E Following value must be specified: ***VALUE_0*****.**

Explanation: Cannot proceed. A required value is missing in the KMIP message.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP0016E Parsed object is null.

Explanation: Cannot proceed. No object is in the KMIP message payload.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP0101E Attribute name *VALUE_0* **not recognized.**

Explanation: Cannot parse. An unrecognized attribute name is in the KMIP message.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP0102E Attribute value of attribute *VALUE_0* **is not a single field.**

Explanation: Cannot parse. Received multiple values for single-valued attribute in a KMIP message.

System action: The requested operation fails.

Administrator response: See the KMIP specification for details. Correct the input and retry the operation.

CTGKP0103E Attribute value *VALUE_0* **is not of type** *VALUE_1*.

Explanation: Cannot parse. The data type of the attribute value is incorrect.

System action: The requested operation fails.

Administrator response: See the KMIP specification for more information. Correct the input and retry the operation.

CTGKP0105E *VALUE_0* **for attribute** *VALUE_1* **is not of type** *VALUE_2*.

Explanation: Cannot parse. The data type of the attribute value is incorrect. See the KMIP specification for more information.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP0106E KMIPDataStructure is not a primitive attribute.

Explanation: Cannot parse. The data type of the attribute is incorrect.

System action: The requested operation fails.

Administrator response: See the KMIP specification for more information. Correct the input and retry the operation.

CTGKP0107E At least one field must be specified for ObtainUsageAllocation.

Explanation: All fields cannot be null for ObtainUsageAllocation operation. Incorrect parameters were received.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP0108E Either bytes or object values must be specified.

Explanation: Combination of input parameters received for obtaining Usage Allocation operation is incorrect. Either bytes or object values must be specified.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP0301E No credential structure found inside the authentication structure.

Explanation: Unexpected error. Asked for credential but the authentication structure did not contain credentials. This might occur because of incorrect input parameters.

System action: The requested operation fails.

Administrator response: Check the audit logs. Correct the input and retry the operation.

CTGKP0302E Cryptographic algorithm not specified and not contained within key value.

Explanation: Cannot parse the message. Cryptographic algorithm cannot be determined.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP0303E Cryptographic length not specified and not contained within key value.

Explanation: Cannot parse the message. Cryptographic length cannot be determined.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP0304E Key value received but key format type not defined.

Explanation: Cannot parse the message. The required value for key format attribute is not specified.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP0305E Data type *VALUE_0* not valid for field key value.

Explanation: Cannot parse the message. An incorrect type for key value is specified. Expected OCTET_STRING data type.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP0306E Tag value for the field *VALUE_0* is incorrect.

Explanation: Cannot parse the message. A tag value is not correct for the specified field.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP0307E Vendor extension key value *VALUE_0* not supported.

Explanation: Cannot parse the message. The value for the specified field is not correct.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP0308E Transparent key format type *VALUE_0* not recognized.

Explanation: Cannot parse the message. The value for the specified field is not correct.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP0309E One of the following must be present: (Private Exponent), (P and Q) or (Prime Exponent P and Prime Exponent Q).

Explanation: Cannot parse the message. The value for the RSA private key is not correct.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP0310E Value not valid for the parameter *VALUE_0*, received *VALUE_1*.

Explanation: The parameter value in the message is either not specified or not valid.

System action: The requested operation failed.

Administrator response: Correct the input and retry the operation.

CTGKP0311E Transparent symmetric key not specified.

Explanation: The parameter value in the request is not specified or not valid.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP0401E End of file reached. No more bytes to read.

Explanation: Cannot read the message completely. No more bytes are available.

System action: The requested operation fails.

Administrator response: Check the audit logs. Correct the input and retry the operation.

CTGKP0402E Error reading *VALUE_0* bytes.

Explanation: Cannot read the message completely because the expected number of bytes cannot be read.

System action: The requested operation fails.

Administrator response: Check the audit logs. Correct the input and retry the operation.

CTGKP0403E Number of bytes to read *VALUE_0* is more than maximum size *VALUE_1*.

Explanation: Cannot read the message because the value for bytes is not valid.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP0404E Tried to skip *VALUE_0* bytes, could only skip *VALUE_1*.

Explanation: Cannot read the message because there are not enough bytes to skip.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP0405E Maximum level of nesting reached *VALUE_0*.

Explanation: Cannot parse the message. Reached the maximum level of nesting.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP0406E Unknown type *VALUE_0*.

Explanation: Cannot parse the message because an object in the message is an unknown type.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP0407E For the object *VALUE_0*, received length of *VALUE_1* that is not valid.

Explanation: Cannot parse the message. The length of the data type is not valid for the specified object type.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP0408E Structure length *VALUE_0* is not a multiple of 8.

Explanation: Cannot parse the message because the length of the structure is not a multiple of 8.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP0409E Structure length *VALUE_0* is a negative number.

Explanation: Cannot parse the message because the length of the structure is not a valid number. Expected a positive number that is a multiple of 8.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP0410E Padding length *VALUE_0* bytes is more than maximum size *VALUE_1* bytes.

Explanation: Cannot parse the message because padding length received is not valid.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP0411E *VALUE_0* length of *VALUE_1* bytes is more than maximum size *VALUE_2* bytes.

Explanation: Cannot parse the message because the length of the object in the message is not valid.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP0601E KMIPDataTypeObject is null.

Explanation: Internal error. Cannot encode the message, received null value for KMIPDataTypeObject.

System action: The requested operation fails.

Administrator response: Check the audit logs. Correct the input and retry the operation.

CTGKP0602E Structure length *VALUE_0* is not a multiple of *VALUE_1* .

Explanation: Cannot encode the message. The length of the structure is not valid.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP0603E Unknown type *VALUE_0*.

Explanation: Cannot encode the message. The type of the object in the message is unknown.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP0701E KLMAdapter classname is null.

Explanation: Internal error. KLMAdapter classname should be set with installation.

System action: The requested operation fails.

Administrator response: Try restarting the server. If the probem continues, you might need to contact IBM Support.

CTGKP0702E Only request messages can be processed.

Explanation: Received the message with a type other than request.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation

CTGKP0703E The method processBatchItem returned null.

Explanation: Internal error. Cannot proceed because an unexpected error ocurred.

System action: The requested operation fails.

Administrator response: Check the audit logs. Correct the input and retry.

CTGKP0704E Can not authenticate the client.

Explanation: Did not receive a client certificate for authentication. The client certificate is required to be sent on SSL communication on operations other than Query.

System action: The request fails.

Administrator response: Make sure a client sends a certificate that SKLM trusts and retry the request.

CTGKP0801E Certificate type *VALUE_0* **not supported.**

Explanation: Cannot proceed because a value in the message is not valid.

System action: The requested operation fails.

Administrator response: Check the audit logs. Correct the input and retry.

CTGKP0802E Key representation not supported.

Explanation: Cannot proceed because a value in the message is not valid.

System action: The requested operation fails.

Administrator response: Check the audit logs. Correct the input and retry.

CTGKP0803E Could not create managed object.

Explanation: Cannot proceed. An internal error occurred.

System action: The requested operation fails.

Administrator response: Check the audit logs. Correct the input and retry.

CTGKP0804E Key format type must be opaque for secret data objects. Value not valid: *VALUE_0*

Explanation: Cannot proceed. A value in the message is not valid.

System action: The requested operation fails.

Administrator response: Check the audit logs. Correct the input and retry.

CTGKP0805E Template must have at least one attribute.

Explanation: Cannot proceed. The template must have at least one attribute.

System action: The requested operation fails.

Administrator response: Check the audit logs. Correct the input and retry.

CTGKP0901E Error constructing KMIP Attribute object. Exception is: *VALUE_0*

Explanation: Cannot proceed. An internal error occurred.

System action: The requested operation fails.

Administrator response: Check the audit logs. Correct the input and retry.

CTGKP0902E Error getting KMIPConfig object.

Explanation: Cannot proceed. An internal error occurred.

System action: The requested operation fails.

Administrator response: Check the audit logs. Correct the input and retry.

CTGKP0903E Error in KMIP Attribute object. Exception is: *VALUE_0*

Explanation: Cannot proceed. An internal error occurred.

System action: The requested operation fails.

Administrator response: Check the audit logs. Correct the input and retry.

CTGKP0904E Unique identifier of the *VALUE_0* is null.

Explanation: Cannot proceed. An unexpected error occurred.

System action: The requested operation fails.

Administrator response: Check the audit logs. Correct the input and retry.

CTGKP0905E Must specify either Common Key Specification, or both Private and Public Key Specifications.

Explanation: Cannot proceed. The input is not valid.

System action: The requested operation fails.

Administrator response: Check the audit logs. Correct the input and retry.

CTGKP0906E More than two unique identifiers specified.

Explanation: Cannot proceed. Received a value that is not valid.

System action: The requested operation fails.

Administrator response: Check the audit logs. Correct the input and retry.

CTGKP0907E No templates or attributes specified.

Explanation: Cannot process this message. This operation requires either a template name or at least one attribute.

System action: The requested operation fails.

Administrator response: Check the audit logs. Correct the input and retry.

CTGKP0908E *VALUE_0* payloads do not exist.

Explanation: Cannot proceed. An unexpected error occurred.

System action: The requested operation fails.

Administrator response: Check the audit logs. Correct the input and retry.

CTGKP0909E Unable to create KMIP message object.

Explanation: Cannot proceed. An unexpected error occurred.

System action: The requested operation fails.

Administrator response: Check the audit logs. Correct the input and retry.

CTGKP0910E At most, two unique identifiers can be specified.

Explanation: Cannot proceed. Received a value that is not valid.

System action: The requested operation fails.

Administrator response: Check the audit logs. Correct the input and retry.

CTGKP0911E Error in KMIP Attribute object. Exception is *VALUE_0*

Explanation: Cannot proceed. Received a value that is not valid.

System action: The requested operation fails.

Administrator response: Check the audit logs. Correct the problem and retry.

CTGKP0912E No attribute names returned.

Explanation: Cannot proceed. An internal error occurred.

System action: The requested operation fails.

Administrator response: Check the audit logs. Correct the problem and retry.

CTGKP0913E At least one attribute name must be requested.

Explanation: Cannot proceed. The input is not valid.

System action: The requested operation fails.

Administrator response: Check the audit logs. Correct the input and retry.

CTGKP0914E Error in ManagedObject.

Explanation: Internal error. Received null or not valid value for ManagedObject.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP0915E Object tag does not match the type of cryptographic object returned.

Explanation: Internal error. The tag does not match the type of the object returned.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP0916E Either byte or object count must be specified.

Explanation: Cannot process the GetUsageAllocation request. Specify either byte or object count.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP0917E No attributes specified for Locate request.

Explanation: Cannot process Locate request. Specify at least one attribute.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP0918E Error constructing KMIP Attribute object. Exception is *VALUE_0*.

Explanation: Internal error. Cannot process the request.

System action: The requested operation fails.

Administrator response: Check the audit logs. Correct the input and retry the operation.

CTGKP0919E No attributes specified on *VALUE_0*.

Explanation: Cannot process the request. Specify at least one attribute.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP0920E Multiple values for field *VALUE_0* received.

Explanation: Cannot process the request. Received multiple values of the field. However, the field is single-valued.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP0921E Unrecognized field *VALUE_0*.

Explanation: Cannot process the request. The field is not recognized.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP0922E Received null object in a Put request.

Explanation: Cannot process the Put request. There must be an object in the message.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP0923E Replaced unique identifier must be specified if Put function is Replace.

Explanation: Cannot process the Put request.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP0924E Error constructing managed object. Error is *VALUE_0*.

Explanation: Cannot process the Put request. The input is not valid.

System action: The requested operation fails.

Administrator response: Check the audit logs. Correct the input and retry the operation.

CTGKP0925E Error constructing KMIP Attribute Object. Error is *VALUE_0*.

Explanation: Cannot process the request. An internal error occurred.

System action: The requested operation fails.

Administrator response: Check the audit logs. Correct the input and retry the operation.

CTGKP0926E *VALUE_0* not specified.

Explanation: Cannot process the Put request. One of the required fields is not specified.

System action: The requested operation fails.

Administrator response: Check the audit logs. Correct the input and retry the operation.

CTGKP0927E All Query functions are null.

Explanation: Cannot process the Query request. Specify at least one Query function.

System action: The requested operation fails.

Administrator response: Check the audit logs. Correct the input and retry the operation.

CTGKP0928E Unique Identifier of the object is null.

Explanation: Internal error. Could not get a response to send back to the client. Cannot process the request.

System action: The requested operation fails.

Administrator response: Check the audit logs. Correct the input and retry the operation. If the problem persists, call IBM Support.

CTGKP0929E If Revocation Reason is Compromised, Compromise OccurrenceDate must be specified.

Explanation: Cannot process the request. The input is not valid.

System action: The requested operation fails.

Administrator response: Check the audit logs. Correct the input and retry the operation.

CTGKP0930E Compromise Occurrence Date not specified and Revocation Reason is Compromised.

Explanation: Internal error. Cannot process the request. The input might not be valid.

System action: The requested operation fails.

Administrator response: Check the audit logs. Correct the input and retry the operation.

CTGKP0931E Error in constructing date-time object while processing *VALUE_0*. **Error is** *VALUE_0*

Explanation: Cannot process the request because one of the KMIP attributes cannot be constructed from the specified input.

System action: The requested operation fails.

Administrator response: Check the audit logs. Correct the input and retry the operation.

CTGKP0932E Error in Revoke request fields. Error is *VALUE_0*

Explanation: Cannot process the request. The input is not valid.

System action: The requested operation fails.

Administrator response: Check the audit logs. Correct the input and retry the operation.

CTGKP0933E Certificates and Unique Identifiers are both null, nothing to validate.

Explanation: Cannot process the request. The input is not valid.

System action: The requested operation fails.

Administrator response: Check the audit logs. Correct the input and retry the operation.

CTGKP0934E Error constructing KMIP Certificate object. Error is *VALUE_0*

Explanation: Cannot process the request. The input is not valid.

System action: The requested operation fails.

Administrator response: Check the audit logs. Correct the input and retry the operation.

CTGKP0935E Certificates and Unique Identifiers are both null.

Explanation: Cannot process the request. The input is not valid.

System action: The requested operation fails.

Administrator response: Check the audit logs. Correct the input and retry the operation.

CTGKP1001E *VALUE_0* **is null.**

Explanation: Internal error occurred. One of the required objects is null. SSL initialization failed.

System action: KMIP SSL Listener is not available to accept KMIP requests.

Administrator response: Check the audit logs. If you intend to use KMIP, ensure that your SSL configuration properties are defined properly and that the KMIP SSL port does not conflict with the port numbers that other applications use. If the problem persists, call IBM Support.

CTGKP1005E Error initializing KMIP Servlet.

Explanation: Internal error occurred. There is a problem in initialization.

System action: Server is not able to accept KMIP requests.

Administrator response: Check the audit logs. Correct the problem and retry. If the problem persists, call IBM Support.

CTGKP1007E KMIP is supported on SSL protocol only.

Explanation: KMIP supports SSL protocol only. Http(s) is not supprted.

System action: The request fails.

Administrator response: Retry sending the request using SSL protocol.

CTGKP2001E Element size *VALUE_0* bytes is larger than maximum size *VALUE_1* bytes.

Explanation: Cannot parse the message. The size is not valid.

System action: Cannot process the request.

Administrator response: Check the audit logs. Correct the input and retry the operation.

CTGKP2002E Unable to read the response.

Explanation: Cannot send the response to the client. An internal error occurred.

System action: Cannot process the request.

Administrator response: Check the audit logs. Correct the problem and retry. If the problem persists, call IBM Support.

CTGKP2003E Byte array to send as a response message is null.

Explanation: Cannot send the response to the client. An internal error occurred.

System action: Cannot process the request.

Administrator response: Check the audit logs. Correct the problem and retry. If the problem persists, call IBM Support.

CTGKP2004E Unable to connect to server. Error is *VALUE_0*

Explanation: Cannot connect to the server.

System action: Cannot send the request to the server.

Administrator response: Check the audit logs. Make sure the hostname and the port number of the server is correctly specified and retry. If the problem persists, call IBM Support.

CTGKP3001E Multiple values for *VALUE_0* field received.

Explanation: Multiple values received for the field, which is not a multi-valued attribute. Specify a single value for this field.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP3002E Message is null.

Explanation: Cannot parse the message. The input is not valid.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP3003E Message is not a response. Message is *VALUE_0*

Explanation: Cannot parse the message. The message type is not valid.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP3004E No payload in batch item.

Explanation: Cannot parse the message. The input is not valid.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP3005E More than one batch item in the message.

Explanation: Cannot process the message. The input is not valid.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP3006E Field in *VALUE_0* is unknown *VALUE_1*.

Explanation: Cannot parse object. Received an unrecognized field.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP3007E Can only get response payload from response messages.

Explanation: Internal error. Tried to get response payload from a request message.

System action: The requested operation fails.

Administrator response: Check the audit logs. Take appropriate action and retry.

CTGKP3008E Result status is pending but there are no batch items in the message *VALUE_0*.

Explanation: Internal error. The result status is not correct when no batch items remain.

System action: The requested operation fails.

Administrator response: Check the audit logs. Take appropriate action and retry.

CTGKP3009E The result status is unknown in this message *VALUE_0*.

Explanation: Internal error. The result status is not correct. Cannot send the response to the client.

System action: The requested operation fails.

Administrator response: Check the audit logs. Take appropriate action and retry.

CTGKP3010E Error reading result status from this response *VALUE_0*.

Explanation: Internal error. The result status is not correct when no batch items remain. Cannot send the response to the client.

System action: The requested operation fails.

Administrator response: Check the audit logs. Take appropriate action and retry.

CTGKP3011E No batch items in the message.

Explanation: Cannot process the message. Received no batch items.

System action: The requested operation fails.

Administrator response: Check the audit logs. Take appropriate action and retry.

CTGKP3012E The result status cannot be recognized.

Explanation: Internal error. Received a result status that is not valid. Cannot send the response to the client.

System action: The requested operation fails.

Administrator response: Check the audit logs. Take appropriate action and retry.

CTGKP3013E Error occurred while creating KMIPCheckException. Error is *VALUE_0*.

Explanation: Internal error occurred. Cannot process KMIP Check operation.

System action: The requested operation fails.

Administrator response: Check the audit logs. Take appropriate action and retry.

CTGKP3014E Operation failed. Error is *VALUE_0*.

Explanation: Internal error occurred.

System action: The requested operation fails.

Administrator response: Check the audit logs. Take appropriate action and retry.

CTGKP3015E The result status cannot be recognized.

Explanation: Internal error. Cannot send the response to the client because the result status is not correct.

System action: The requested operation fails.

Administrator response: Check the audit logs. Take appropriate action and retry.

CTGKP3016E Cryptographic algorithm is null.

Explanation: No value for the cryptographic algorithm in the message.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP3017E Year must be 4 characters.

Explanation: Cannot process the message. The value for the year field is not valid.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP3018E Month must be between 1 and 12.

Explanation: Cannot process the message. The value for the month field is not valid.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP3019E Day must be between 1 and 31.

Explanation: Cannot process the message. The value for the day field is not valid.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP3020E Hour must be between 0 and 24.

Explanation: Cannot process the message. The value for the hour field is not valid.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP3021E Minute must be between 0 and 59.

Explanation: Cannot process the message. The value for the minute field is not valid.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP3022E Second must be between 0 and 59.

Explanation: Cannot process the message. The value for the second field is not valid.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP3023E Unknown cryptographic algorithm *VALUE_0*

Explanation: Cannot process the message. A cryptographic algorithm parameter that is not valid.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP3024E Unsupported key format *VALUE_0*

Explanation: Cannot process the message. The key format parameter is not valid.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP3025E Wrapping method *VALUE_0* **is not supported.**

Explanation: Cannot process the message. The wrapping method in the message is not supported.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP3026E The MAC/Signature verification failed.

Explanation: Cannot process the message. The signature verification failed.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP3027E Block cipher mode *VALUE_0* **only supported for AES algorithm.**

Explanation: Cannot process the message. The block cipher mode is not valid.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP3028E Block cipher mode is null.

Explanation: Cannot process the message. The block cipher mode is missing in the message.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP3029E Block cipher mode *VALUE_0* **only supported for algorithm** *VALUE_1*.

Explanation: The block cipher mode in the message is not supported for this algorithm.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP3030E The key value type *VALUE_0* **of the encryption key is not supported.**

Explanation: Cannot process the message. The input is not valid.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP3031E The wrapping key itself cannot already be wrapped.

Explanation: Cannot process the message. The wrapping key is not valid.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP3032E Unique Identifier mismatch: query does not match response.

Explanation: Internal error. The unique identifier in the query and response do not match.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP3033E No Last Changed Date attribute for specified cryptographic object.

Explanation: Operation fails because the specified cryptogrpahic object does not have the last changed date attribute.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP3034E Parameter *VALUE_0* is not a Boolean value.

Explanation: Cannot process the message. Specify the parameter as a Boolean value.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP3035E Parameter *VALUE_0* not defined.

Explanation: Cannot process the message because the parameter is not specified.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP3036E Parameter *VALUE_0* is empty.

Explanation: Cannot process the message because the parameter is not specified.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP4001E Zero or more than one attributes in container.

Explanation: Internal error. Cannot get single-valued attribute because there are zero or more than one attribute present in the container.

System action: The requested operation fails.

Administrator response: Check the audit logs. Correct the input and retry the operation.

CTGKP4002E Template name cannot be null.

Explanation: Cannot process the message. An unexpected error occurred.

System action: The requested operation fails.

Administrator response: Check the audit logs. Correct the input and retry the operation.

CTGKP4003E Error adding *VALUE_0* attribute. Error is *VALUE_1*.

Explanation: Cannot process the message. An internal error occurred.

System action: The requested operation fails.

Administrator response: Check the audit logs. Correct the input and retry the operation.

CTGKP4004E The index can only be set when there is exactly one attribute in the container.

Explanation: Cannot process the message. An unexpected error occurred.

System action: The requested operation fails.

Administrator response: Check the audit logs. Correct the input and retry the operation.

CTGKP4005E No index specified for *VALUE_0* attribute value and index zero already used.

Explanation: Cannot process the message. The parameters in the message are incorrect.

System action: The requested operation fails.

Administrator response: Check the audit logs. Correct the input and retry the operation.

CTGKP4006E This method cannot be batched with other requests.

Explanation: Cannot process the message. The input is not valid.

System action: The requested operation fails.

Administrator response: Check the audit logs. Correct the input and retry the operation.

CTGKP4007E Length of Wrapping Key ID array must be the same as the array for unique identifiers.

Explanation: Cannot process the message. An internal error occurred.

System action: The requested operation fails.

Administrator response: Check the audit logs. Correct the input and retry the operation.

CTGKP4008E List of unique identifiers is null.

Explanation: Cannot process the message. The input parameter is not valid.

System action: The requested operation fails.

Administrator response: Check the audit logs. Correct the input and retry the operation.

CTGKP4009E Response was asynchronous although the Asynchronous Indicator is false.

Explanation: Cannot process the message. The input parameter is not valid.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP4010E Thread.sleep() interrupted. Error is *VALUE_0*

Explanation: Cannot process the message. An internal error occurred.

System action: The requested operation fails.

Administrator response: Check the audit logs. Take appropriate action and retry. If the problem persists, call IBM Support.

CTGKP4011E No templates or attributes specified.

Explanation: Cannot process a message. The input parameter is not valid.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP4012E Attribute name is null.

Explanation: Cannot process a message. The input is not valid.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP4013E Attribute value is null.

Explanation: Cannot process a message. An internal error occurred.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP4014E Unexpected response payload received.

Explanation: Cannot process a message. An internal error occurred.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP4015E Error parsing returned object in the response.

Explanation: Cannot process a message. An internal error occurred.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP4016E Error getting attributes from the response.

Explanation: Cannot process a message. An internal error occurred.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP4017E No attributes in the message.

Explanation: Cannot process a message. An internal error occurred.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP4018E Unique identifier in the response does not match the one in the request.

Explanation: Cannot process a message. An internal error occurred.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP4019E Asynchronous correlation value does not match. Result was *VALUE_0*.

Explanation: Cannot process a message. An internal error occurred.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP4020E Error encoding request message *VALUE_0*.

Explanation: Cannot process a message. An internal error occurred.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP4021E Reply is not a KMIP response.

Explanation: Cannot process a message. An internal error occurred.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP4022E The response is empty even though batching is not used.

Explanation: Cannot process a message. An internal error occurred.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP4023E Error while wrapping object. Error is *VALUE_0*.

Explanation: Cannot process a message. An internal error occurred.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP4024E Error while unwrapping object. Error is *VALUE_0*.

Explanation: Cannot process a message. An internal error occurred.

System action: The requested operation fails.

Administrator response: Correct the input and retry the operation.

CTGKP4025E Error connecting to the remote server. Error is *VALUE_0*.

Explanation: Cannot process a message, unexpected error occurred.

System action: The requested operation fails.

Administrator response: Ensure that the remote server is up and running.

Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785 U.S.A.

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law :

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement might not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
2Z4A/101
11400 Burnet Road
Austin, TX 78758 U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_. All rights reserved.

If you are viewing this information in softcopy form, the photographs and color illustrations might not be displayed.

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 “Copyright and trademark information” at 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