CHAPTER 6

License Inventory Management Functions

This chapter provides information about the following license inventory management functions:

annotate_licenses

Synopsis

```
annotate_licenses ($token, [@lic_ids], [@annotation])
```

Description

This function allows you to annotate a license with comments that you provide.

This function does not return till the call completes. Such a call is called a blocking call. It returns a Cisco::CLM::Common::IDStatus object containing the status of the operation.

Input Parameters

Parameter	Type	Value	Description
token	UserToken, mandatory	—	Token that represents your authorization pass, which is obtained after you invoke the login function and are authenticated by the back-end server.
lic_ids	Array of string, mandatory	Up to 256 ASCII characters in the range from x21 to x7A	ID of License objects.
annotation	Array of string, mandatory	Text string of up to 99 characters	Text of the annotation for each license. Cisco License Manager does not check the length of the annotation parameter. Cisco IOS software truncates the text if it exceeds the character limit.

Return

The function returns a Cisco::CLM::Common::IDStatus on completion.

Error and Exception

If a system error prevents the operation from completing, a RemoteException is thrown.

deploy_licenses

Synopsis

```
deploy_licenses ($token, [@lic_ids])
```

Description

This function deploys the given licenses to their target devices.

This function does not return till the call completes. Such a call is called a blocking call. It returns a Cisco::CLM::Common::IDStatus object containing the staus of the operation.

Input Parameters

Parameter	Type	Value	Description
token	UserToken, mandatory	—	Token that represents your authorization pass, which is obtained after you invoke the login function and are authenticated by the back-end server.
lic_ids	Array of string, mandatory	Up to 256 ASCII characters in the range from x21 to x7A	Array of license ID.

Return

The function returns a Cisco::CLM::Common::IDStatus on completion.

Error and Exception

If a system error prevents the operation from completing, a RemoteException is thrown.

get_device_ids_with_undeployed_licenses

Synopsis

```
get_device_ids_with_undeployed_licenses ($token, $group)
```

Description

This function returns the IDs of devices that have undeployed licenses.

Input Parameters

Parameter	Type	Value	Description
token	UserToken, mandatory	—	Token that represents your authorization pass, which is obtained after you invoke the login function and are authenticated by the back-end server.
group	String, mandatory	Up to 64 ASCII characters in the range from x21 to x7A	Name of the device group (or null, if looking for ungrouped devices).

Return

This function returns the Cisco::CLM::Common::IDStatus object for instances of IDStatusItem array that contain device IDs.

Error and Exception

If a system error prevents the operation from completing, a RemoteException is thrown.

When an error occurs, the Status object returns the none SUCCESS error code and an error message.

get_pak_ids_with_undeployed_licenses

Synopsis

```
get_pak_ids_with_undeployed_licenses($token,$group)
```

Description

This function returns the IDs of PAKs that have undeployed licenses.

Input Parameters

Parameter	Type	Value	Description
token	UserToken, mandatory	—	Token that represents your authorization pass, which is obtained after you invoke the login function and are authenticated by the back-end server.
group	String, mandatory	Up to 64 ASCII characters in the range from x21 to x7A	Name of the PAK group (or null, if looking for ungrouped devices)

Return

This function returns the Cisco::CLM::Common::IDStatus object for instances of IDStatusItem array that contain PAK IDs.

Error and Exception

If a system error prevents the operation from completing, a RemoteException is thrown.

When an error occurs, the Status object returns the none SUCCESS error code and an error message.

get_licenses_on_device

Synopsis

```
get_licenses_on_device ($token, $dev_id)
```

Description

This function retrieves license information that resides on a given device.

Input Parameters

Parameter	Type	Value	Description
token	UserToken, mandatory	—	Token that represents your authorization pass, which is obtained after you invoke the login function and are authenticated by the back-end server.
dev_ids	String, mandatory	Up to 256 ASCII characters in the range from x21 to x7A	Device ID string.

Return

This function returns a Cisco::CLM::Common::LicenseStatus object.

Error and Exception

If a system error prevents the operation from completing, a RemoteException is thrown.

When an error occurs for an element in the input array, the returned status object contains an error code and error message.

get_rehostable_skus_by_device

Synopsis

```
get_rehostable_skus_by_device ($token, $dev_id)
```

Description

This function retrieves an array of SKUs that can be used for license rehost for a specified input device. This function will in turn trigger a request to the Software Infrastructure and Fulfillment Technology (SWIFT) group to query all SKUs whose licenses have been deployed on the device.

Input Parameters

Parameter	Type	Value	Description
token	UserToken, mandatory	—	Token that represents your authorization pass, which is obtained after you invoke the login function and are authenticated by the back-end server.
dev_id	String, mandatory	Up to 64 ASCII characters in the range from x21 to x7A	ID of the device.

Return

This function returns a Cisco::CLM::Common::RehostableSKUStatus object. If no licenses have been obtained for the device, the RehostableSKU field in the returned RehostableSKUStatus is set to null.

Error and Exception

If a system error prevents the operation from completing, a RemoteException is thrown.

When error occurs, the information is contained in the returned status object.

get_rehost_info

Synopsis

```
get_rehost_info ($token, [@dev_ids])
```

Description

This function returns the RehostInfo of each given device. Each RehostInfo contains a rehost request and a permission ticket or a rehost ticket.

Input Parameters

Parameter	Type	Value	Description
token	UserToken, mandatory	—	Token that represents your authorization pass, which is obtained after you invoke the login function and are authenticated by the back-end server.
dev_id	String Array, mandatory	Up to 256 ASCII characters in the range from x21 to x7A	Array of device IDs.

Return

This function returns a Cisco::CLM::Common::RehostInfoStatus object.

Error and Exception

If a system error prevents the operation from completing, a RemoteException is thrown.

If an operation error occurs, the RehostInfoStatus object contains the none SUCCESS error code and error message. Otherwise, you must traverse the RehostInfoStatusItem array to retrieve all of the RehostInfo objects.

init_rehost_license

Synopsis

```
init_rehost_license ($token, $rehost_req)
```

Description

The limitation of rehosting from the Cisco Product License Registration Portal is that there can be only one PermissionTicket acquired per device until a new license is obtained. This means that there is only one PermissionTicket and one RehostTicket per device at any time.

This function is the first step of the rehost process. The process consists of several steps, including getting a permission ticket from the Cisco Product License Registration Portal, retrieving the rehost ticket from the device, sending the rehost ticket to the Cisco Product License Registration Portal to obtain the license, and deploying the license to the destination device.

The obtained PermissionTicket is stored in local storage and is later used to revoke the license from the source device.

Input Parameters

Parameter	Type	Value	Description
token	UserToken, mandatory	—	Token that represents your authorization pass, which is obtained after you invoke the login function and are authenticated by the back-end server.
rehost_req	RehostRequest, mandatory	—	An object that represents the request.

Return

This function returns the Cisco::CLM::Common::Status object.

Error and Exception

If a system error prevents the operation from completing, a RemoteException is thrown.

Status contains an error code and an error message if the operation is not successful.

list_all_licenses_in_pak

Synopsis

```
list_all_licenses_in_pak ($token, $pak_id)
```

Description

This function returns an array of license IDs that belong to a given PAK.

Input Parameters

Parameter	Type	Value	Description
token	UserToken, mandatory	—	Token that represents your authorization pass, which is obtained after you invoke the login function and are authenticated by the back-end server.
pak_id	String, mandatory	Up to 64 ASCII characters in the range from x21 to x7A	PAK ID that contains the licenses.

Return

This function returns a string array of License ID contained by the PAK. If the pak_id is invalid, this function returns null.

Error and Exception

If a system error prevents the operation from completing, a RemoteException is thrown.

When an error occurs, this function returns null.

obtain_license

Synopsis

```
obtain_license ($token, [@lic_req], $deploy)
```

Description

This function downloads the information that is associated with a given product authorization key (PAK) IDs from the Cisco Product License Registration Portal and stores the information in the inventory. The first function only obtains the licenses; the second function obtains the licenses and deploys them.

This function does not return till the call completes. Such a call is called a blocking call. It returns a Cisco::CLM::Common::IDStatus object containing the staus of the operation.

Input Parameters

Parameter	Type	Value	Description
token	UserToken, mandatory	—	Token that represents your authorization pass, which is obtained after you invoke the login function and are authenticated by the back-end server.
lic_reqs	Array of LicenseRequest, mandatory	—	Array of LicenseRequest.
deploy	Boolean, mandatory	True, False	True to ask the server to deploy all licenses obtained.

Return

The function returns a Cisco::CLM::Common::IDStatus on completion.

Error and Exception

If a system error prevents the operation from completing, a RemoteException is thrown.

obtain_license_for_rehost

Synopsis

```
obtain_license_for_rehost ($token, $rehost_req)
```

Description

The limitation of rehosting from the Cisco Product License Registration Portal is that there can be only one PermissionTicket acquired per device until a new license is obtained. This means that there is only one PermissionTicket and one RehostTicket per device at any time.

This function is the third step of the rehost process. The process consists of several steps, including getting a permission ticket from the Cisco Product License Registration Portal, retrieving the rehost ticket from the device, sending the rehost ticket to the Cisco Product License Registration Portal to obtain the license, and deploying the license to the destination device.

Input Parameters

Parameter	Type	Value	Description
token	UserToken, mandatory	—	Token that represents your authorization pass, which is obtained after you invoke the login function and are authenticated by the back-end server.
rehost_req	RehostRequest, mandatory	—	An object that represents the request.

Return

This function returns a Cisco::CLM::Common::LicenseStatus object.

Error and Exception

If a system error prevents the operation from completing, a RemoteException is thrown.

The Status object contains an error code and error message if an operation error occurs. Otherwise, you must traverse the LicenseStatusItem array to retrieve all of the license objects.

re_obtain_license

Synopsis

```
re_obtain_license ($token, $dev_id)
```

Description

This function requests that the Cisco Product License Registration Portal resend a license. After licenses are received, it updates and synchronizes Cisco License Manager data storage. It does not deploy licenses to a device.

Input Parameters

Parameter	Type	Value	Description
token	UserToken, mandatory	—	Token that represents your authorization pass, which is obtained after you invoke the login function and are authenticated by the back-end server.
dev_id	String	Up to 64 ASCII characters in the range from x21 to x7A	ID of device to which to resend the license.

Return

This function returns a Cisco::CLM::Common::IDStatus object.

Error and Exception

If a system error prevents the operation from completing, a RemoteException is thrown.

When an error occurs, the information is contained in the returned Status object.

read_licenses

Synopsis

```
read_licenses ($token, [@lic_ids])
```

Description

This function retrieves an array of license objects from the inventory using the given device IDs.

Input Parameters

Parameter	Type	Value	Description
token	UserToken, mandatory	—	Token that represents your authorization pass, which is obtained after you invoke the login function and are authenticated by the back-end server.
lic_ids	Array of string, mandatory	Up to 256 ASCII characters in the range from x21 to x7A	Array of License ID.

Return

This function returns a Cisco::CLM::Common::LicenseStatus object.

Error and Exception

If a system error prevents the operation from completing, a RemoteException is thrown.

When an error occurs on an element in the input array, a LicenseStatus object is returned with information about the error.

rehost_license

Synopsis

```
rehost_license ($token, $rehost_req)
```

Description

This function sends requests to rehost licenses from one device to another. This process contains several steps, including retrieving a permission ticket from the Cisco Product License Registration Portal, retrieving a rehost ticket from the source device, and sending the rehost ticket to the Cisco Product License Registration Portal to obtain new licenses for the destination device and deploy the new licenses to the destination device. These steps are encapsulated by this function as a single operation. The obtained license is stored in local storage and can be used later for deployment to the destination devices.

Input Parameters

Parameter	Type	Value	Description
token	UserToken, mandatory	—	Token that represents your authorization pass, which is obtained after you invoke the login function and are authenticated by the back-end server.
rehost_req	RehostRequest, mandatory	—	Object that represents the request.

Return

This function returns a Cisco::CLM::Common::Status object, which contains the error code and message. If the operation is successful, the ClmErrors.SUCCESS error code is returned. If it is unsuccessful, the none ClmErrors.SUCCESS error code and the error message are returned.

Error and Exception

If a system error prevents the operation from completing, a RemoteException is thrown.

When an error occurs, this function returns a Status object that contains the error code and error message.

resend_license

Synopsis

```
resend_license ($token, $dev_id)
```

Description

This function resends licenses to a device to restore corrupted license files. The function requests all licenses that have been obtained from the Cisco Product License Registration Portal, saves them into the License Manager database, and then reploys them to the device.

Input Parameters

Parameter	Type	Value	Description
token	UserToken, mandatory	—	Token that represents your authorization pass, which is obtained after you invoke the login function and are authenticated by the back-end server.
dev_id	String, mandatory	Up to 64 ASCII characters in the range from x21 to x7A	ID of device to which to resend the license.

Return

This function returns a Cisco::CLM::Common::Status object, which contains the error code and error message. If the operation is successful, the ClmErrors.SUCCESS error code is returned.

Error and Exception

If a system error prevents the operation from completing, a RemoteException is thrown.

When an error occurs, this function returns false.

revoke_license_for_rehost

Synopsis

```
revoke_license_for_rehost ($token, $rehost_req)
```

Description

This function is used when rehost fails in the middle of a task. The limitation of rehosting from the Cisco Product License Registration Portal is that there can be only one PermissionTicket acquired per device until a new license has been obtained. This means that there is only one PermissionTicket and one RehostTicket per device at any time.

This function is the second step of the rehost process. The process consists of several steps, including retriving a permission ticket from the Cisco Product License Registration Portal, retrieving the rehost ticket from the device, sending the rehost ticket to the Cisco Product License Registration Portal to obtain the license, and deploying the license to the destination device.

The obtained PermissionTicket is stored in local storage and is later used to revoke the license from the source device. It is removed if the revoke operation is successful, and the RehostTicket is stored in local storage for thenext step of the rehost process.

Input Parameters

Parameter	Type	Value	Description
token	UserToken, mandatory	—	Token that represents your authorization pass, which is obtained after you invoke the login function and are authenticated by the back-end server.
rehost_req	RehostRequest, mandatory	—	Object that represents the request.

Return

This function returns a Cisco::CLM::Common::Status object.

Error and Exception

If a system error prevents the operation from completing, a RemoteException is thrown.

Status contains an error code and error message if the operation is not successful.

transfer_rma_device_licenses

Synopsis

```
transfer_rma_device_licenses($token,$source_dev_udi,$dest_dev_udi,$deploy)
```

Description

This function transfers the licenses from a return material authorization (RMA) device to a new device. If the Boolean deploy value is set to true, the Cisco License Manager server deploys the licenses to the new device.

Input Parameters

Parameter	Type	Value	Description
token	UserToken, mandatory	—	Token that represents the user's authorization pass, which is obtained after the user invokes the login function and gets authenticated by the back-end server.
Source_dev_udi	String		UDI of the RMA device.
Dest_dev_udi	String		UDI of the target device.
deploy	Boolean		True if licenses are to be deployed to the target device.

Return

This function returns the Status object.

Error and Exception

If a system error prevents the operation from completing, a RemoteException is thrown.

write_licenses

Synopsis

```
write_licenses ($token, [@lics])
```

Description

This function writes the given license objects into the inventory. The input license objects can be existing instances of license retrieved from the inventory by the function read_licenses.

Input Parameters

Parameter	Type	Value	Description
token	UserToken, mandatory	—	Token that represents your authorization pass, which is obtained after you invoke the login function and are authenticated by the back-end server.
lics	Array of License, mandatory	Up to 256 ASCII characters in the range from x21 to x7A	Array of License objects.

Return

This function returns a Cisco::CLM::Common::IDStatus object.

Error and Exception

If a system error prevents the operation from completing, a RemoteException is thrown.

When an error occurs on an element in the input array, a status object is returned with information about the error.