



CHAPTER 4

Device Management Functions

This chapter provides information about the following device management functions:

- [create_device_by_ip_addr](#), page 4-1
- [create_devices_by_udi](#), page 4-3
- [delete_devices](#), page 4-3
- [discover_devices](#), page 4-5
- [get_active_rma_devices](#), page 4-6
- [get_allowed_operation_by_device_platform](#), page 4-4
- [get_discovery_settings](#), page 4-7
- [get_discovery_transports](#), page 4-8
- [get_transport_method_by_device_platform](#), page 4-8
- [list_all_devices_in_group](#), page 4-9
- [list_all_groups_by_device](#), page 4-10
- [list_applicable_devices_by_sku](#), page 4-10
- [list_device_ids_by_filter](#), page 4-11
- [poll_device_license_info](#), page 4-12
- [read_devices](#), page 4-12
- [write_devices](#), page 4-13

create_device_by_ip_addr

Synopsis

```
create_device_by_ip_addr ($token, $ip, $group, $dev_auth_info, [@transports])
```

Description

This function creates device objects in the inventory using a given IP address and associates devices with a specified group.

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.
ip	Array of string, mandatory	IP address	IP addresses. Each address is used for creating a device object.
group	String, mandatory	Up to 64 ASCII characters in the range from x21 to x7A	Name of the group.
dev_auth_info	Cisco::CLM::Common::DiscoveryAuthInfo mandatory	—	DeviceAuthentication is to be used when communicating with discovered devices to get license information. DeviceAuthentication contains username, password, and enable password.
transports	Cisco::CLM::Common::Device::TransportMethod array, mandatory	HTTP, HTTPS, TELNET, SSH	This parameter specifies which connection transport method is used for each IP entry. Cisco License Manager assumes that the caller knows the connection method used for each device.

Return

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

The following example shows error codes and messages from DeviceStatus object:

```
# The general error code of the operation.
my $err_code = $device_status->get_error_code;
# The general error message of the operation.
my $err_msg = $device_status->get_error_message;
# A list of status for each individual element in the
# bulk operation.
my $device_status_items = $id_status->get_device_status_items;
# Iterate through the list to get individual status.
foreach my $k (@{$device_status_items}){
# Get the individual Device object returned by the operation.
my $device = $k->get_device;
#Get the individual error code corresponding to
# the Device
my $item_err_code = $k->get_error_code;
# Get the individual error message corresponding to
# the Device.
my $item_err_msg = $k->get_error_message;
}
```

Error and Exception

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

When an operation error occurs, a DeviceStatus object is returned with information about the error. To inspect the individual element status, you must traverse the DeviceStatusItem array within DeviceStatus. Each DeviceStatusItem contains the Device object, error code, and error message.

create_devices_by_udi

Synopsis

```
create_devices_by_udi($token, [@udis], [$group])
create_devices_by_udi($token, [@udis])
```

Description

This function creates device objects in the inventory using a given unique device identifier (UDI).

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.
udis	Array of string, mandatory	Up to 64 ASCII characters in the range from x21 to x7A	Array of UDIs. Each UDI is used for creating a device object.
group	String, optional	Up to 64 ASCII characters in the range from x21 to x7A	Name of the device group.

Return

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

Error and Exception

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

When an operation error occurs, a DeviceStatus object is returned with information about the error. To inspect the individual element status, you must traverse the DeviceStatusItem array within DeviceStatus. Each DeviceStatusItem contains the Device object, error code, and error message.

delete_devices

Synopsis

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

Description

This function deletes device objects from the inventory using the given device IDs. If a device that is being deleted belongs to a group, it is removed from the group.

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	Array of string, mandatory	Up to 64 ASCII characters in the range from x21 to x7A	Array of device IDs.

Return

This function returns 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 IDstatus object is returned with information about the error.

get_allowed_operation_by_device_platform

Synopsis

```
get_allowed_operation_by_device_platform ($token, $platform)
```

Description

This function returns an array of allowed license operations for a specified platform.

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.
platform	Cisco::CLM::Common::Device::DevicePlatform mandatory	CISCO12K, CSL, IPS, UNKNOWN, URLF, WNBU	Device platform.

Return

The function returns an array of `Cisco::CLM::Common::Device::LicenseOperation` for a given device platform. If the given device platform is unknown, it will return an empty array.

Error and Exception

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

discover_devices

Synopsis

```
discover_devices($token, $subnet,$subnet_mask, $group, $dev_auth_info, [@transports],
[@policy_ids] )
```

Description

This function generates an inventory of devices that are discovered in a given subnet and adds the devices to a specified device group.

The Cisco License Manager currently supports Cisco IOS Telnet and secure shell (SSH) for non-agent-enabled device discovery, thus prolonging the discovery time. If the network devices are configured the same way, you can provide the transport method (such as HTTP, Telnet, or SSH), which speeds up the discovery and reduces the discovery process time. If a null or empty string array is passed in, Cisco License Manager uses all supported methods to discover the devices.

This function does not return till the call completes, that is, the script is executed only after the discovery is complete. Such a call is called a blocking call. It returns a `Cisco::CLM::Common::IDStatus` object containing the status of the operation.

This function supports an additional parameter, `policy_ids`, which is an array of policy IDs that are run after completion of the discovery.

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.
subnet	String, mandatory	Network address	Subnet to conduct the auto-discovery.
subnet_mask	String, mandatory	—	Subnet mask.
group	String, mandatory	Up to 64 ASCII characters in the range from x21 to x7A	Group to which the discovered devices should be added.

Parameter	Type	Value	Description
dev_auth_info	DiscoveryAuthInfo, mandatory	—	<p>A DiscoveryAuthInfo is used when communicating with discovered devices to retrieve license information. If no authentication is needed for the devices in the network, this parameter can be set to null.</p> <p>DiscoveryAuthInfo contains an array of username and password pairs and an array of enable passwords. When Cisco License Manager tries to communicate with the device, the instances of DiscoveryAuthInfo are tried one at a time until they can be authenticated by the device. If Telnet is used, there are only three tries.</p>
transport	Cisco::CLM::Common::Device::TransportMethod array, mandatory	HTTP, TELNET, SSH	<p>If the devices in a subnet are set up with the same device discovery method, the device discovery process is speeded up because Cisco License Manager uses only the specified method to quickly determine if the device license is supported. This parameter specifies which connection transport is used. Cisco License Manager uses the specified transports by entry order.</p> <p>If null or zero length is entered, Cisco License Manager uses all available methods to discover the device.</p>
policy_ids	Array of string		Array of policy IDs that runs after completion of the discovery.

Return

This function returns a Cisco::CLM::Common::IDStatus object containing the status of the operation.

Error and Exception

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

get_active_rma_devices

Synopsis

```
get_active_rma_devices($token)
```

Description

This function returns a list of connected devices marked as RMA (meaning return material authorization) in the CLM server storage.

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.

Return

The function returns DeviceStatus, in which Cisco::CLM::Common::DeviceStatus contains the Cisco::CLM::Common::Device device objects and error messages.

Error and Exception

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

get_discovery_settings

Synopsis

```
get_discovery_settings ($token)
```

Description

This function retrieves all discovery information after Cisco License Manager Release 2.2 is installed. DiscoverySetting is a data structure in data storage, which stores the network IP, network mask, TransportMethods and DiscoveryAuthInfo of each discovery operation.

Input Parameters

Parameter	Type	Value	Description
token	UserToken, mandatory	—	Token that represents the administrator user's authorization pass, which is obtained after the user invokes the login function and gets authenticated by the back-end server.

Return

The function returns the Cisco::CLM::Common::DiscoverySetting array, which is available only to an administrator user. If you are not an administrator user, the function returns a null value. The function returns an empty array if no discovery operation was invoked before this function call.

A sample program:

```
my $discovery_setting = license_manager->get_discovery_settings($token);
foreach my $info (@{$discovery_setting})
{
    my $discovery_auth_info = $info->get_discovery_auth_info;
    my $ip_address = $info->get_net_ip_addr;
    my $net_mask = $info->get_net_mask;
}
```

Error and Exception

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

get_discovery_transports

Synopsis

```
get_discovery_transports ($token)
```

Description

This function returns an array of Cisco::CLM::Common::Device::Transport Method set by the administrator user during Discovery. Only a user with administrator privileges can run this API.

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.

Return

This function returns an array of transport methods (Cisco::CLM::Common::Device::TransportMethod) available for the specified platform.

Error and Exception

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

get_transport_method_by_device_platform

Synopsis

```
get_transport_method_by_device_platform ($token, $platform)
```

Description

This function returns an array of transport methods available for a specified platform.

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.
platform	Cisco::CLM::Common::Device::DevicePlatform mandatory	CSL, WNBU, URLF, CISCO12K, IPS, UNKNOWN	The name of the platform can be CSL, WNBU, URLF, CISCO12K, IPS, or UNKNOWN.

Return

This function returns an array of platforms (Cisco::CLM::Common::Device::TransportMethod) available for a specified platform.

Error and Exception

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

list_all_devices_in_group

Synopsis

```
list_all_devices_in_group ($token, $group)
```

Description

This function returns an array of device IDs that belong to a given device group. If the group is null, this function returns an array of device IDs that do not belong to any device group.

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 the function is looking for ungrouped devices.

Return

This function returns an array of device IDs if the group is valid. If the group is invalid or empty, an array of size zero is returned.

Error and Exception

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

list_all_groups_by_device

Synopsis

```
list_all_groups_by_device ($token, $dev_id)
```

Description

This function returns an array of groups to which a given device belongs.

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	Device ID.

Return

This function returns an array of groups if the device is valid. If the device is invalid or belongs to no group, an array of size zero 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 null.

list_applicable_devices_by_sku

Synopsis

```
list_applicable_devices_by_sku ($token, $sku, $group)
```

Description

This function returns an array of device objects that are applicable for the given SKU (stock keeping unit).

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.
sku	SKU, mandatory	—	SKU object.
group	String	—	Group name.

Return

The function returns an array of Cisco::CLM::Common::Device matching the platforms of a SKU in a given group. If group argument is null, matching devices in all groups will be returned.

Error and Exception

If sku is null, this API returns null. If a platform in a SKU is null or UNKNOWN, or no device matches the SKU platform, this API returns an array of zero length.

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

list_device_ids_by_filter

Synopsis

```
list_device_ids_by_filter($token, $device_type, $device_model, [@features])
```

Description

This function returns a list of device IDs that contains the device_type, device_model, and license feature.

Each filter can be set to null for no filtering. If all the filters are set to null, the operation is treated as an error, and null will be returned.

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.
device_type	String, optional	—	String that represents device type.
device_model	String, optional	—	String that represents device model.
features	Array of string, optional	—	Array of string that represent the license feature names.

Return

This function returns an array of string. Each string is a device ID.

Error and Exception

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

When an error occurs, this function returns null. An array length of zero means that no device is found.

poll_device_license_info

Synopsis

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

Description

This function communicates with the given devices, retrieving license information from them and storing the information in the inventory.

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.
dev_ids	Array of string, mandatory	Up to 64 ASCII characters in the range from x21 to x7A	Array of device IDs.

Return

This function returns a `Cisco::CLM::Common::IDStatus` object containing the status of the operation.

Error and Exception

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

read_devices

Synopsis

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

Description

This function uses the given device ID to retrieve an array of device objects from the inventory.

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	Array of string, mandatory	Up to 64 ASCII characters in the range from x21 to x7A	Array of device IDs.

Return

This function returns a `Cisco::CLM::Common::DeviceStatus` object.

Error and Exception

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

When an operation error occurs, a `DeviceStatus` object is returned with information about the error. To inspect the individual element status, you must traverse the `DeviceStatusItem` array within `DeviceStatus`. Each `DeviceStatusItem` contains the Device object, error code, and error message.

write_devices

Synopsis

```
write_devices ($token, [@devices])
```

Description

This function writes the given device objects into the inventory. The input device objects can be new instances of devices returned by the `create_devices` function or existing ones retrieved from the inventory by the `read_devices` function.

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.
devices	Array of device, mandatory	—	Array of device objects, which can be new instances of a device class or existing ones retrieved from the inventory by the <code>readDevice()</code> function.

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.