



# CHAPTER 10

## PAK Management Functions

---

This chapter provides information about the following product authorization key (PAK) management functions:

- [create\\_paks](#), page 10-1
- [delete\\_paks](#), page 10-2
- [download\\_pak\\_info](#), page 10-3
- [list\\_all\\_folders\\_by\\_pak](#), page 10-4
- [list\\_all\\_paks\\_in\\_folder](#), page 10-4
- [list\\_pak\\_contain\\_features](#), page 10-5
- [read\\_paks](#), page 10-6
- [write\\_paks](#), page 10-6

### create\_paks

#### Synopsis

```
create_paks ($token, [@pak_ids], $folder)
create_paks ($token, [@paks], $folder)
create_paks ($token, [@pak_ids])
create_paks ($token, [@paks])
```

#### Description

This function creates PAK objects in the inventory. The input array of `pak_ids` is used as the name for each instance of the PAK object. If the input parameter `folder` is provided, the newly created PAK is added to the folder.

The third and fourth forms of the function take the array of PAK objects as the input parameter. These two functions allow you to create a PAK object and fill in the attributes.

**Input Parameters**

Parameter	Type	Value	Description
token	UserToken, mandatory	—	A token that represents the user's authorization pass, which is obtained after the user invokes the login function and is authenticated by the back-end server.
pak_ids	Array of string, mandatory	ID is a string containing up to 64 ASCII characters in the range from x21 to x7A	An array of PAK IDs, used as Name for each instance of PAK object.
folder	String, optional	Name is a string containing up to 64 ASCII characters in the range from x21 to x7A	The name of the folder. <b>Note</b> Some forms of this function do not require a folder name.
paks	Array of PAK, mandatory	PAK object	An array of PAK objects.

**Return**

This function returns a Cisco::CLM::Common:: PAKStatus 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.

## delete\_paks

**Synopsis**

```
delete_paks($token, [@pak_ids])
```

**Description**

This function deletes PAK objects from the inventory using the given PAK IDs. If a PAK belongs to a folder, it is removed from the folder.

**Input Parameters**

Parameter	Type	Value	Description
token	UserToken, mandatory	—	A token that represents your authorization pass, which is obtained after you invoke the login function and are authenticated by the back-end server.
pak_ids	Array of string, mandatory	ID is a string containing up to 64 ASCII characters in the range from x21 to x7A	An array of PAK IDs.

**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.

## download\_pak\_info

**Synopsis**

```
download_pak_info ($token, [@pak_ids])
```

**Description**

This function downloads the information that is associated with the given PAK IDs from the Cisco Product License Registration Portal and stores the information in the inventory.

This function blocks until the call completes. It returns a `Cisco::CLM::Common::IDStatus` object containing the status of the operation.

**Input Parameters**

Parameter	Type	Value	Description
token	UserToken, mandatory	—	A token that represents your authorization pass, which is obtained after you invoke the login function and are authenticated by the back-end server.
pak_ids	Array of String, mandatory	ID is a string containing up to 64 ASCII characters in the range from x21 to x7A	An array of PAK IDs that identify interested PAKs. A PAK ID is generated by using the <code>createPAK</code> function, passing a PAK name as a parameter to get the PAK 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.

## list\_all\_folders\_by\_pak

**Synopsis**

```
list_all_folders_by_pak($token, $pak_id)
```

**Description**

This function returns a list of all folders to which the PAK belongs.

**Input Parameters**

Parameter	Type	Value	Description
token	UserToken, mandatory	—	A 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	ID is a string containing up to 64 ASCII characters in the range from x21 to x7A	PAK ID, which is used as the ID for each instance of the PAK object.

**Return**

This function returns a string array of folder names to which the given PAK belongs. If the PAK does not exist, 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\_all\_paks\_in\_folder

**Synopsis**

```
list_all_paks_in_folder($token, $folder)
```

**Description**

This function returns an array of PAK IDs that belong to the given folder. If the input folder is null, this function returns an array of PAK IDs that do not belong to any folder.

**Input Parameters**

Parameter	Type	Value	Description
token	UserToken, mandatory	—	A token that represents your authorization pass, which is obtained after you invoke the login function and are authenticated by the back-end server.
folder	String, mandatory	Name is a string containing up to 64 ASCII characters in the range from x21 to x7A	Name of the PAK folder, or null if the function is looking for ungrouped PAKs.

**Return**

This function returns an array of string containing PAK IDs that belong to the folder. If the folder does not exist or is empty, 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\_pak\_contain\_features

**Synopsis**

```
list_pak_contain_features($token, [@features])
```

**Description**

This function returns a list of PAKs that contains all the license features in the given feature list.

**Input Parameters**

Parameter	Type	Value	Description
token	UserToken, mandatory	—	A token that represents your authorization pass, which is obtained after you invoke the login function and are authenticated by the back-end server.
features	Array of String, mandatory	Features is a string containing up to 64 ASCII characters in the range from x21 to x7A	An array of string that represents the license feature names. The array of string is an AND relationship.

**Return**

This function returns an array of string. Each string is a PAK 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 no PAK was found.

## read\_paks

**Synopsis**

```
read_paks($token, [@pak_ids])
```

**Description**

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

**Input Parameters**

Parameter	Type	Value	Description
token	UserToken, mandatory	—	A token that represents your authorization pass, which is obtained after you invoke the login function and are authenticated by the back-end server.
pak_ids	Array of String, mandatory	ID is a string containing up to 64 ASCII characters in the range from x21 to x7A	An array of PAK IDs.

**Return**

This function returns a Cisco::CLM::Common::PAKStatus 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.

## write\_paks

**Synopsis**

```
write_paks($token, [@pak])
```

**Description**

This function writes the given PAK objects into the inventory. The input PAK objects can be new instances of PAK returned by the create\_paks function or existing ones retrieved from the inventory by the read\_paks function.

**Input Parameters**

Parameter	Type	Value	Description
token	UserToken, mandatory	—	A token that represents your authorization pass, which is obtained after you invoke the login function and are authenticated by the back-end server.
paks	Array of PAK, mandatory	PAK object	An array of PAK 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.

