



CHAPTER 11

PAK Management Functions

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

- [asyncDownloadPAKInfo](#), page 11-2
- [createPAKs](#), page 11-3
- [deletePAKs](#), page 11-3
- [listAllFoldersByPAK](#), page 11-4
- [listAllPAKsInFolder](#), page 11-5
- [listPAKContainFeatures](#), page 11-6
- [readPAKs](#), page 11-6
- [writePAKs](#), page 11-7

asyncDownloadPAKInfo

Synopsis

```
public String asyncDownloadPAKInfo(UserToken token, String jobGroup, String[] pakIDs,
IDStatusListener listener) throws RemoteException;
```

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 is nonblocking and returns a request ID to the caller immediately. When calling this function, the client program provides a listener object that implements the StatusListener interface. When the function is completed, the onStatus() method in the listener object is invoked.

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.
jobGroup	String, mandatory	—	A group to which the asynchronous job belongs. If it is set to null, the returned job ID will be used as job group name.
pakIDs	Array of string, mandatory	—	Array of PAK IDs.
listener	IDStatusListener object, mandatory	—	Object that implements IDStatusListener interface.

Return

This function returns a request ID to the caller. When the operation completes, the status is provided as the input parameter of the onStatus() method. The listener receives the IDStatus upon operation complete. The IDStatus contains a list of IDStatusItem, which includes the PAK ID, the error code, and the error message of the operation of this PAK ID.

Error and Exception

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

createPAKs

Synopsis

```
public PAKStatus createPAKs(UserToken token, String[] pakNames, String folder) throws
RemoteException;
```

```
public PAKStatus createPAKs(UserToken token, PAK[] paks, String folder) throws
RemoteException;
```

Description

This function creates PAK objects in the inventory in the given folder. If folder is null, the newly created PAK is added to the Default folder.

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.
pakNames	String, mandatory	Up to 64 ASCII characters in the range from x21 to x7A	Array of PAK objects.
folder	String, mandatory	—	Name of the folder.
paks	Array of PAK, mandatory	PAK object	Array of PAK objects.

Return

This function returns PAKStatus objects. Cisco License Manager generates a PAK ID in the returned PAK object. You must use this ID in readPAKs and deletePAKs. When there is non-system error for an element in the input array, the information is contained in the returned status object.

Error and Exception

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

deletePAKs

Synopsis

```
public IDStatus deletePAKs(UserToken token, String[] pakIDs) throws RemoteException;
```

Description

This function deletes PAK objects from the inventory by using the given PAK IDs. If the PAK belongs to any folder, it is removed from the folder. The PAK ID is obtained from createPAKs or listAllPAKs.

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.
pakIDs	Array of string, mandatory	—	Array of PAK IDs.

Return

This function returns IDStatus objects. If there is an operation error, the error code and the error message are returned in the status object.

Error and Exception

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

listAllFoldersByPAK

Synopsis

```
public String[] listAllFoldersByPAK(UserToken token, String pakID) throws RemoteException;
```

Description

This function returns an array of all folders to which a PAK 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.
pakID	Array of string, mandatory	—	Array of PAK IDs.

Return

This function returns an array of folders if the PAK is valid. If the PAK is invalid, a array of zero size is returned.

Error and Exception

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

listAllPAKsInFolder

Synopsis

```
public IDPagingInfo listAllPAKsInFolder(UserToken token, String folder, Pagination
pageinfo) throws RemoteException;
```

Description

This function returns IDPagingInfo object, which includes status, total count, and an array of PAK IDs belonging to a given folder. If the folder is null, the function returns PAK IDs in the Default folder.

If a valid offset and a max are entered, pagination is provided. For example, if the result set has 100 records, and you want 10 records at a time, you can request one page at a time by specifying the offset and the max in Pagination:

To get first page:

```
Pagination p = new Pagination(0, 10)
```

To get fifth page:

```
Pagination p = new Pagination(50, 10)
```

To get all records without paging:

```
Pagination p = new Pagination(0, -1)
```

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.
folder	String, mandatory	—	Name of the PAK folder, or null if the function is looking for the Default folder.
pageinfo	DataObject, mandatory	—	Pagination object includes an offset and a max. Offset specifies the offset relative to the first record set to return. Max specifies the maximum number of records to return. If the maximum is set to -1, all records are returned.

Return

This function returns IDPagingInfo object, which includes an array of PAK IDs. If the offset is larger than the number of records, it returns an array of zero size. If there is an operation error, the Status object contains an error code and a message.

Error and Exception

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

listPAKContainFeatures

Synopsis

```
public String[] listPAKContainFeatures(UserToken token, String[] features) throws
RemoteException;
```

Description

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

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.
features	Array of string, mandatory	—	String array that represents the license feature names. The API lists those PAKs that contain all the features mentioned in the array of features (AND relationship).

Return

This function returns a string array. Each string is a PAK ID.

Error and Exception

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

readPAKs

Synopsis

```
public PAKStatus readPAKs(UserToken token, String[] pakIDs) throws RemoteException;
```

Description

This function retrieves an array of PAK objects from the inventory by using the given PAK IDs. The PAK IDs are obtained from the createPAKs or the listAllPAKs 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.
pakIDs	Array of string, mandatory	—	Array of PAK IDs.

Return

This function returns PAKStatus objects. If there is operation error, the Status object contains the error code and the message.

Error and Exception

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

writePAKs

Synopsis

```
public IDStatus writePAKs(UserToken token, PAK[] paks) throws RemoteException;
```

Description

This function writes the given PAK objects into the inventory. The input PAK objects can be new instances of PAK returned by the createPAKs() function or existing ones retrieved from the inventory by the readPAKs() function or the or listAllPAKs 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.
paks	Array of PAK, mandatory	PAK object	Array of PAK objects.

Return

This function returns IDStatus objects. If there is operation error, the Status object contains the error code and the message.

Error and Exception

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

