CHAPTER 8

License Line Management Functions

This chapter provides information about these license line management functions:

asyncAnnotateLicenseLines

Synopsis

```
public String asyncAnnotateLicenseLines(UserToken token, String jobGroup, String[]
liclineIDs, String[] annotation, IDStatusListener listener) throws RemoteException;
```

Description

This function allows you to annotate license lines with comments.

This function is nonblocking and returns a request ID to the caller immediately. While calling this function, the client program provides a listener object that implements StatusListener interface. When the operation is complete, 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	—	—	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.

Parameter	Type	Value	Description
liclineIDs	Array of string, mandatory	—	Array of license line IDs.
annotation	Array of string, mandatory	Text string of up to 99 characters	Text of the annotation for each license line. Cisco License Manager does not check the length of the annotation parameter. Cisco IOS software returns an error if the character limit is exceeded.
listener	IDStatusListener object, mandatory	—	Object that implements IDStatusListener interface.

Return

This function returns a request ID string to the caller. When the operation is complete, the status is provided as the input parameter of onStatus() method. The listener will receive the IDStatus, which includes a list of IDStatusItem. Each IDStatusItem contains the license line ID, error code, and the error message of the operation.

Error and Exception

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

asyncDeployLicenseLines

Synopsis

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

Description

This function deploys the given license lines to their target devices.

This function is nonblocking and returns a request ID to the caller immediately. While calling this function, the client program provides a listener object that implements the StatusListener interface. When the function is complete, 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	—	—	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.

Parameter	Type	Value	Description
liclineIDs	Array of string, mandatory	—	Array of license line IDs to be deployed.
listener	IDStatusListener object, mandatory	—	Object that implements IDStatusListener interface.

Return

This function returns a request ID string to the caller. When the operation is complete, the status is provided as the input parameter of onStatus() method. The listener will receive the IDStatus, which includes a list of IDStatusItem. Each IDStatusItem contains the license line ID, error code, and the error message of the operation.

Error and Exception

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

getLicenseLinesByLicense

Synopsis

```
public LicLineStatus getLicenseLinesByLicense(UserToken token, String licID) throws
RemoteException;
```

Description

This function retrieves license lines that are contained in a given license.

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.
licID	String, mandatory	—	License ID.

Return

This function returns LicLineStatus objects. If there is an 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.

getLicenseLinesOnDevice

Synopsis

```
public LicLineStatus getLicenseLinesOnDevice(UserToken token, String devID) throws
RemoteException;
```

Description

This function retrieves license lines that are contained in the given device ID.

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.
devID	String, mandatory	—	Device ID.

Return

This function returns LicLineStatus objects. If there is an 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.

listExpiredLicenseLines

Synopsis

```
public HashSet<Expired License> listExpiredLicenseLines (UserToken token) throws
RemoteException;
```

Description

This function returns a list of license lines that have expired.

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 HashSet objects. If there is an operation error, it returns null.

Error and Exception

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

listExpiringLicenses

Synopsis

```
public ExpiringLicenseStatus listExpiringLicenses (UserToken token, int dayToExpire)
throws RemoteException;
```

Description

This function returns a list of licenses that will expire in the specified number of days.

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.
dayToExpire	int, mandatory	Range from 1 to 100	Number of days left for licenses to expire.

Return

This function returns ExpiringLicenseStatus, which contains the ExpiringLicenseStatusItem array, error number, and the error message. If an operation error occurs, the error number returns the none SUCCESS. If SUCCESS is returned, you must go through each LicenseLineStatusItem to retrieve all LicenseLine objects. Expired Licenses are not included in the list.

Error and Exception

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

readLicenseLines

Synopsis

```
public LicLineStatus readLicenseLines(UserToken token, String[] licLineIDs) throws
RemoteException;
```

Description

This function retrieves an array of license line objects from the inventory by using the given license line 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.
licLineIDs	Array of string, mandatory	—	Array of license line IDs.

Return

This function returns LicLineStatus 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.