



CISCO CONFIDENTIAL

CHAPTER 34

Using the Licensing APIs

The CWCS Licensing API allows you to:

- Install and update licenses.
- Retrieve license information.
- Access FLEXlm utilities that can be used to implement a wide variety of licensing models.

The following topics describe how to use the CWCS Licensing APIs with your application:

- [Understanding CWCS Licensing APIs](#)
- [Integrating CWCS Licensing APIs](#)

For basic information on the CWCS Licensing API, see the “[About CWCS Licensing](#)” section on [page 29-6](#).

For more information on CWCS Licensing APIs, see:

- [Licensing Requirements Document, EDCS-295207](#)
- [MDC CORE License Model, EDCS 153881](#)
- [CMF 2.3 System Function Specifications, EDCS-283137](#)
- [CMF Licensing Framework Functional Specification, EDCS-295256](#)

Understanding CWCS Licensing

Software licenses are used to prevent unauthorized use of software products. Historical use of licensing technology in CWCS predecessor products included:

- **CMF-style licensing**, used in products such as RME and Campus Manager. This form of licensing uses proprietary technology and supports evaluation and purchased licenses, but did not address common requirements, such as feature- or size-based licensing.
- **FLEXlm-based licensing**, designed for VMS applications. This form of licensing was implemented in Core 1.0 (and supported in CMF up to version 2.2), but is inappropriate for general use, since it was designed specifically for VMS bundle use cases.

The current CWCS licensing:

- Continues support for the licensing components that existed in CMF 2.2. Applications that currently use that component can continue to do so without any changes.

CISCO CONFIDENTIAL

- Provides a facility for upgrade protection. This enables applications that currently employ CMF-style licensing to support upgrade licenses for customers who have a valid, purchased copy of the previous version of the application.
- Provides a framework for FLEXIm-based licensing. The framework supports feature- or size-based licenses through a simple API. Since some applications will have licensing requirements that are not addressed by this API, the framework provides FLEXIm tools that can be used to implement custom licensing schemes.

The CWCS licensing framework supports the following features:

- Licenses for evaluations, purchases, and non-revenue programs.
- Licenses that specify a resource limit, such as the number of devices that can be managed by an application.
- Licenses that grant the right to use an application or feature within an application.
- Temporary use of PIN to validate the use of a feature.
- Repository to store PIN and PAK (Product Authorization Key).
- API to install licenses and to retrieve license information, including PIN/PAK.
- License Administration GUI.
- FLEXIm toolkit.
- Backup for licenses.

License models other than those noted above such as node-locked licenses, floating licenses, counted licenses, etc., will not be directly supported in CMF 3.0. Applications that require such features must use FLEXIm toolkit to implement a licensing model that meets their requirements.

Using Licensing UI

This topic provides information on the end-user interface of the License Information page:

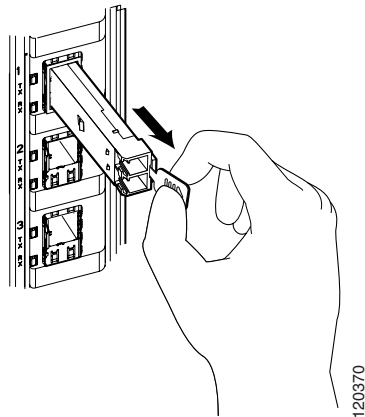
To access the License Information page:

Step 1 In the CiscoWorks Main Page, click **Common Services > Server > Admin**.

The Admin page appears.

Step 2 Click **Licensing** link in the TOC.

The License Information page appears.

CISCO CONFIDENTIAL

The License Information page displays the details of the available licenses.

To update a license:

- Step 1** In the CiscoWorks Main Page, click **Common Services > Server > Admin**.
The Admin page appears.
 - Step 2** Click **Licensing** link in the TOC.
The License Information page appears with the details of available licenses.
 - Step 3** Click **Update**.
The Select License File popup window appears. The CWCS installation directory is displayed in the License File field.
 - Step 4** Click **Browse** to select a license file.
The Server Side File Selector popup window appears.
 - Step 5** Select a drive from the **Drive** list.
 - Step 6** Select the directory from the **Directory Content** list.
 - Step 7** Select the file from the list of files in the directory content area.
The file must be a valid license file with read permissions for the user.
 - Step 8** Click **OK**.
-

CISCO CONFIDENTIAL

Understanding CWCS Licensing APIs

The Licensing framework supports Java interface to install and retrieve license information. The licensing API supports the following operations:

- Retrieve names of all bundles for which a license key/PIN has been installed.
- Install PIN/PAK combination.
- Authorize PIN/PAK associated with an application or a bundled solution.
- Retrieve PIN/PAK associated with an application or a bundled solution.
- Retrieve all installed PIN/PAK.
- Install license keys.
- Authorize license key(s) for an application or a bundled solution.
- Retrieve license data for an application.

License data would contain information such as the installation date, expiration date, license type, number of devices that can be managed, and PAK/PIN or license key(s) associated with the application.

CWCS Licensing Classes

The CWCS Licensing classes are:

- [LicensedFeature](#)
- [LicenseManager](#)
- [LicensePAK](#)

LicensedFeature

LicensedFeature represents a feature or application that has license (such as RME), in addition to a name, version, expiry date and the number of licenses.

Instances of this class contains information on whether the feature has an evaluation, upgrade, purchased, or a special kind of license. Since feature licenses may be delivered via a FLEXlm license file or a PIN, instances of this class may also be queried for the source of licensing information.

The following table provides information on the methods used by LicensedFeature:

Table 34-1 *Methods Used by Licensed Feature*

Method	Description
installTime	Retrieves the time at which the license was installed.
bundleName	Retrieves the name of the bundle that the PAK/PIN licenses.

CISCO CONFIDENTIAL**LicenseManager**

LicenseManager helps applications to manage licenses. Applications based on CMF that need to enforce license restrictions must use this class to install and retrieve licenses.

The following table provides information on the methods used by LicenseManager.

Table 34-2 *Methods Used by LicenseManager*

Method	Description
addLicense	Adds license keys in a FLEXlm license file.
getFeature	Gets information on all versions of a licensed feature or the specified version of a licensed feature. Returns null if no license for the feature has been installed.
getAllFeatures	Gets information on all licensed features. Returns null if no license is installed.
addPAK	Adds a PAK/PIN combination.
getPAK	Retrieves all installed PIN/PAK combinations or the registered PIN/PAK for a bundle. Returns null if no PIN/PAKs are installed.

LicensePAK

LicensePAK represents a PAK/PIN combination and the time at which the PAK/PIN was installed.

The arguments for this class are PAK and PIN associated with the license. The installation time of the LicensePAK is set to current time.

The following table provides information on the methods used by LicensePAK.

Table 34-3 *Methods used by LicensePAK*

Method	Description
getPIN	Retrieves the PIN associated with the license.
getPAK	Retrieves the PAK associated with the license.
installTime	Retrieves the time at which the license was installed.
bundleName	Retrieves the name of the bundle that the PAK/PIN licenses.

Error Codes Generated by APIs

The following table describes the error codes generated by the API:

CISCO CONFIDENTIAL**Table 34-4 Licensing Errors**

Error Code	Description
LicenseError.CorruptLicenseFile	Exception to indicate a corrupt license file.
LicenseError.EvalExtension	Exception to indicate an extension of the evaluation period.
LicenseError.ExpiredLicense	Exception to indicate the expiry of a license.
LicenseError.MalformedPIN	Exception to indicate an incorrect PIN.

Integrating CWCS Licensing APIs

This section provides information on integrating the licensing APIs with your application.

JavaDoc

The Javadoc for the License API can be found at following URL.

<http://nmtgre.cisco.com/auto/embueng/License/>

License Installation

Licenses can be installed during the installation of a bundle, or at a later time.

The license installation UI does not appear during the Common Services 3.0 installation. The UI appears only for the first application in the bundle that is installed.

The application install code must verify the existence of a valid license for that application. If a valid license is missing, the application prompts to enter the license at the installation time.

If the first application installation contains a license key that has entries for all the applications in the bundle, subsequent application installations do not have to prompt for licenses.

- The license file being installed is processed in addition to the existing license files.
- If no valid license entry is provided for an application, the application will run in EVAL mode.
- If a PIN is provided for the application, the application will run in NAG mode until a valid license is provided. For more information on NAG mode, see *CMF Licensing Framework Functional Specification, EDCS-295256*
- If an upgrade license is provided, You must provide a Proof-of-Purchase for that application by running the CMF-provided script with appropriate arguments (validate_upgrade.exe).

Version of the FLEXlm libraries : Version 9.2

CISCO CONFIDENTIAL

PAK and PIN

PAK and PIN are processed in pairs—while the PAK carries no information that is useful to CMF, a PIN is associated with a product or an application, and encodes the type of license (evaluation, upgrade, or purchased) the user has for that product.

1. License Key/PIN is invalid—In this case, since there is no meaningful association between the license key or the PAK/PIN combination and a product, the input is rejected and an error message is displayed with the cause for the failure.
2. License Key/PIN indicates an evaluation license—Scenarios that could be potentially destructive if the current installation is run over of an existing installation of the previous version of the product:
 - The user has only an evaluation license of the previous version—The evaluation license represented by the PIN is invalid and the user will find the product unusable after completing installation.
 - The user has a valid purchased license of the previous version of the product—The user can use the latest installed version in evaluation mode. However, the product would become unusable at the end of the evaluation period, unless a purchased license is installed later.
3. License Key/PIN indicates an upgrade from the previous version—When the user has only an evaluation version of the previous version, installing an upgrade license is equivalent to installing an evaluation license, which is not allowed.
4. License Key/PIN indicates a purchased license—Previously installed versions are irrelevant since the user has a valid, purchased license for the current version.

Since cases 2 and 3 can be potentially destructive, the user is warned, that continuing with the installation may render the product unusable, and the user should be given the option to cancel the installation. If the user proceeds with the installation, the License Framework will process all the license keys, PAKs and PINs that were entered by the user. In the case of fully purchased licenses/PINs, the license will be installed and no further processing is required during product installation. In all other cases, the licenses will be maintained in a state where their use is permitted only after authorization by a product or application.

**Note**

It is the responsibility of the product installation to determine whether a license key/PIN presented by the user is to be treated as a valid license and invoke an API that will authorize the license.

Handling Multiple Licenses

Applications may be licensed as part of an application bundle or separately. It is important to determine the effect of installing a license for an application for which a license has already been installed. In the following section, a PIN that is not superseded by the license key it represents is treated the same as the license key. We consider the following cases here:

CISCO CONFIDENTIAL

Scenario	Description
Installing an evaluation license over a previously installed license.	This operation is invalid and the new evaluation license will be ignored.
Installing an upgrade license over an evaluation license.	The upgrade license will supersede the evaluation license. If a fully purchased license for a previous version of the application is installed, the installed license will be upgraded to the newer version of the application. Otherwise, the upgrade license will be considered <i>incomplete</i> and the application that uses this license will function in NAG mode. For more information on Nag mode, see License Requirements Document, EDCS-295207 .
Installing an upgrade license over a fully purchased license	If the upgrade license is for the same or an earlier version of the application, as compared to the existing one, the upgrade license will be ignored. If the upgrade license is for a more recent version of the application, the installed license will be upgraded to the newer version of the application.
Installing a purchased license over an evaluation license.	Purchased license is installed.
Installing a purchased license over a previously installed purchased license.	Increases the device count. Since purchased licenses do not have an expiration date, the only effect of this operation is to increase the device count, if any. For example, if an application had license for 1000 devices, and a new license with a device count of 500 is installed for the application, the number of licenses for that application increases to 1500 devices.

Interpreting PIN

PIN allows a customer to continue using the product and provide information to the licensing component before the actual license key is entered. PIN provides the following information to the licensing component:

- Bundle name
- Applications and their versions in the bundle
- Allowable device limits per application
- Type of installation (Such as New, Upgrade, SEVT, Network Academy)

PIN does not contain more information than the license key. The license key contains all information that the PIN contains, and more.

The PIN alone does not provide any useful information regarding application name, version, and device limit and installation type. Each application code in the PIN should have a mapping file to get it mapped with an application. To enable this functionality, the evaluation license of each application is used. The

CISCO CONFIDENTIAL

evaluation license of each image has the application name, version and associated application code. The application code in license file is matched with the application code in the PIN to get the details about the application.

The PIN is not a substitute for the license key. The main function of the PIN is to allow the customer to continue using the product in the absence of a license key. This is because very often the user installing the product is not the one who purchased the product and obtains the license key. However, the customer has to enter the license key to move the product from the nagging mode to the full function mode.

If the customer does not enter the actual purchased license key within 90 days, and if a PIN is entered at install time, the product will always function in a NAG mode. In NAG mode, after 90 days, if the PAK and PIN are present, NAG messages will start appearing. If the PAK and PIN are not entered, i.e. if the product is in EVAL mode, the product stops functioning after 90 days.

The customer can enter the license key at install time or using the desktop GUI. It is recommended that the customer enters the actual license key rather than the PIN at install time. The PIN is only a fallback mechanism.

PIN Format

The format of the PIN has been designed to encode all the information required by Common Services licensing. A PIN can encode license information for a bundle or an arbitrary collection of applications.

PIN specification:

- A PIN consists of 32 characters with hyphens separating 8 character sequences. Valid characters in a PIN are [2-9][A-Z] with the exception of the characters **I** and **O**.
- The first eight characters describe the characteristics of a bundle. Bundles are viewed as collections of applications. Hence, a PIN representing an arbitrary collection of applications is treated in the same manner as a bundle.

Description of a bundle can be further broken down as follows:

- Characters in position 1-4 name the bundle. If the name of a bundle is only 3 characters long, the unused character position is filled with 9.
- Characters 5-6 encode the bundle version.
- Character 7 encodes the license type (Permanent, Upgrade, etc).
- Character 8 encodes whether the resource counts in the PIN are cumulative or absolute.
- The last character in the PIN encodes the number of applications in the bundle. Since 0 and 1 are not valid characters, the value for this character is 2 + the number of applications.
- Characters starting from position 8, encodes applications in the bundle, with three characters used for each application. Of the three characters used for an application, the first two encode the name of the application and its version. DNMBU marketing will maintain a table of application codes and their mapping to licensed applications. For example, RME 4.0 may be represented by the code 22, DFM 2.0 by the code 23, etc. The third character used for an application encodes the resource count. All characters that are valid in PINs are mapped to preset resource counts; the character that is used to denote the resource count for an application is that whose associated count is closest to (but not less than) the application resource count.
- The character at position 31 encodes a checksum for all the characters described above.
- All unused character positions are filled with valid characters selected at random.

CISCO CONFIDENTIAL

- Since 10 character positions are used to encode bundle information, the number of applications and the checksum, only 22 characters are available to encode the applications in a bundle. This implies that PINs can be used to encode bundles only when they have 7 or less number of licensed applications.

Understanding License Framework

The Licensing framework provides a way for licensing the evaluation version. The FCS image can be used as both evaluation version and purchased version.

Applications that need to use the Licensing framework must get the application mapping code from CWCS marketing team. (The application code is unique for each application and version. For example, RME 4.0 has an application code 22). Each application must get a static evaluation license file from CMF team. (The static license file contains application code map for the application, and device limit for the Evaluation mode).

This static evaluation license file should be part of the applications image. This must be under `<ApplicationImage>/disk1/eval/<AppName.lic>`. This license file is used during install, if the user selects Eval mode or PIN/PAK mode. The purchased license is generated by SWIFT based on the serial number or PAK.

When an application is installed, the framework will check for available license in the repository based on the Application name, version, and Application code.

If a matching entry is found either in license file or PIN, the framework does not prompt for license details. Otherwise, the user is prompted for license details.

The license file or PIN is validated and the installation proceeds.

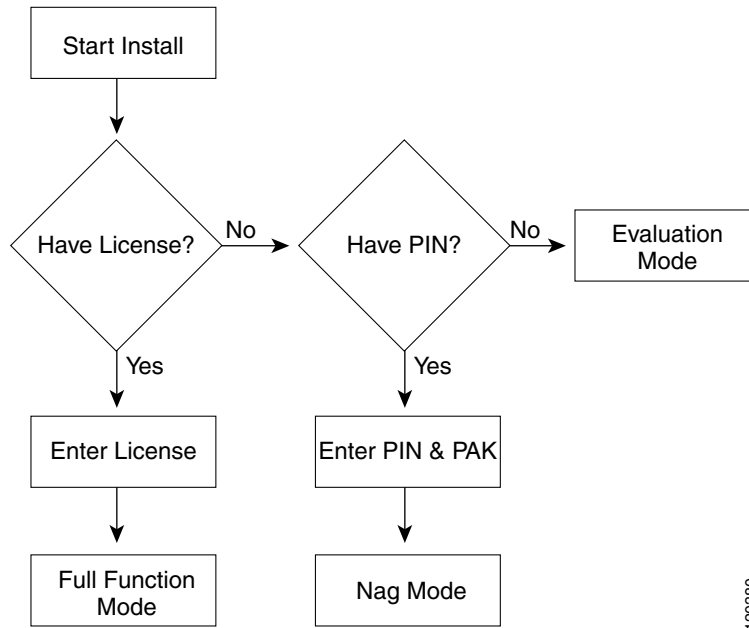
The license repository is `NMSROOT/etc/licenses`. When a new license is added, the information is added to the repository with `.lic` extension.

The PIN/PAK details are stored in `pinpak.data` file.

Flowcharts

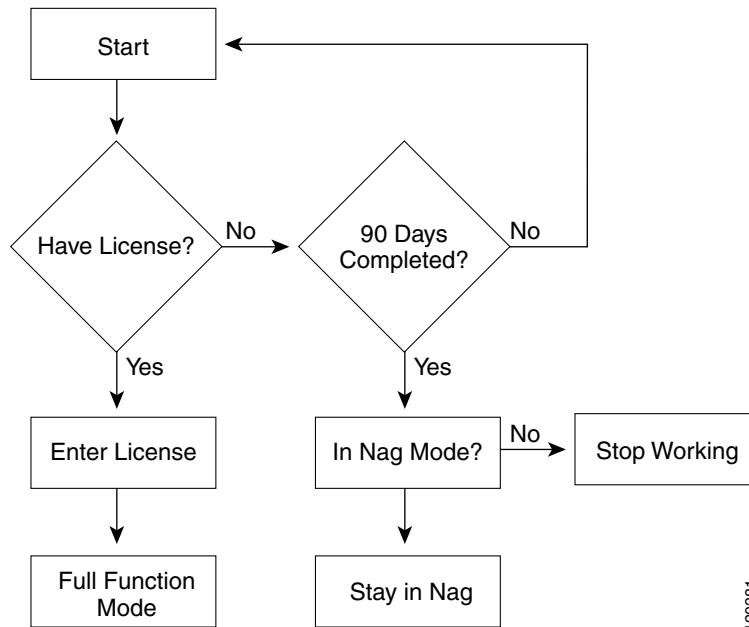
License Key and PIN entry during installation:

CISCO CONFIDENTIAL



120380

License Key and PIN entry within a 90 day period after installation:



120381

CISCO CONFIDENTIAL

Using Licensing Framework With Applications

To use the Licensing Framework provided by CWCS, the application must modify the following sections:

- [Install](#)
- [CW Home Page](#)
- [Runtime Calls](#)

Install

License install framework is part of ITOOLS.

To use the licensing install framework:

-
- Step 1** Create a PRODUCT_INFO TAG in the respective disk.toc with application name and version.

Example: PRODUCT_INFO=rme 4.0



Note Application name must be in lower case.

- Step 2** Get application mapping code for the application from CWCS marketing.

- Step 3** Create an evaluation license for the application and place it under *<ApplicationImage>/disk1/eval/*.

The license file name should read: *<application name>.lic*

CW Home Page

Applications must verify the license type. If the license type indicates that the application is for non-revenue programs, the application should display the message “The product is not licensed for commercial use.”

Examples of non-revenue programs: SEVT, NFR, NA

API Call for getting the license type for an application:

API Call	Description
LicensedFeature.licenseType()	The call returns the type of license for that application.

Runtime Calls

To query licensing details at runtime:

-
- Step 1** Import the following:

```
import com.cisco.nm.license.client.*;
import com.cisco.nm.license.util.*;
```

CISCO CONFIDENTIAL

- Step 2** Create instances of:
- ```
LicenseManager lm = new LicenseManager();
LicensedFeature licFeature = lm.getAllFeatures("Appname", "Version");
```
- Step 3** Use APIs to query the license details at runtime.



**Note** Licensing framework use Log4J for logging. Application using License class should handle logging properly.

**License SDK**

The License framework will include an SDK containing FLEXlm tools and documentation, so that the applications can implement licensing models other than the one supported by CMF. The SDK contains:

- Programs to implement a license server (lmgrd and vendor daemon)
- Runtime libraries and relevant C language header files
- Tools for administering licenses and generating evaluation licenses

**Data Architecture**

Licenses, PINs and PAKs that are installed using the License GUI or API can be retrieved using the License API. No other specifications are made regarding the internal representation of the license data or the mechanisms used to persist such data.

**License File Format**

The license file format specified in [MDC Core License Model Document, EDCS-153881](#) is used for licensing other bundle applications.

In addition to the file format supported in Common Services 2.2 (as described in EDCS-153881), a new license file format will be used to support the features in Common Services 3.0. Applications that intend to use the features described in this document are required to use the new license file format described in section License File Format for Common Services 3.0.

The license file has a fixed structure. Words in **CAPITAL BOLD** are key words, unique to FLEXlm. Words in *Italics* should be replaced by the actual data. Other wording should be there in order for FLEXlm to be able to parse the syntax.

```
VENDOR cisco
INCREMENT licenseInfo cisco 1.0 expirationDate uncounted \
VENDOR_STRING=DeviceLimit HOSTID=Any \
NOTICE=DemoLength SIGN=0
INCREMENT licenseUser cisco 1.0 expirationDate uncounted \
NOTICE=PAK HOSTID=Any SIGN=0
INCREMENT PIX cisco 1.0 expirationDate uncounted \
NOTICE=PAK VENDOR_STRING=DeviceLimit HOSTID=Any SIGN=0
INCREMENT IOS cisco 1.0 expirationDate uncounted \
NOTICE=PAK VENDOR_STRING=DeviceLimit HOSTID=Any SIGN=0
```

**CISCO CONFIDENTIAL**

```

INCREMENT IDS cisco 1.0 expirationDate uncounted \
NOTICE=PAK VENDOR_STRING=DeviceLimit HOSTID=Any SIGN=0
INCREMENT C3K cisco 1.0 expirationDate uncounted \
NOTICE=PAK VENDOR_STRING=DeviceLimit HOSTID=Any SIGN=0

```

The first INCREMENT is followed by the keyword *licenseInfo* to identify that this feature line will describe the general license information.

| Field                 | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|-----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>cisco</i>          | Name of the Vendor.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <i>1.0</i>            | Version. FLEXlm supports supplying versions between 1.0 and 2.0 inclusive. As other versions of CORE come out, we can increment the 1.0 version.                                                                                                                                                                                                                                                                                                                                                                     |
| <i>expirationDate</i> | This can either be the keyword “permanent” to indicate that the license is purchased and never expires, or can be a date to indicate that this is a demo license key. The expiration date is the date on which the license disk becomes invalid and further installations with this license key are invalid. The date format is dd-mmm-yyyy, where dd and yyyy are numeric, and mmm is the three letter abbreviation for the month. For example, Jan 5, 2002 would be represented as 05-jan-2002 and not 5-jan-2002. |
| <i>uncounted</i>      | Required as the customer will not be running a FLEXlm license server.                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <i>DeviceLimit</i>    | The number of devices licensed for an application. If unlimited, use -1.                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| <i>DemoLength</i>     | The number of days a demo license disk is valid for after installation. In case of a purchased license this value is -1 to indicate that the license does not expire.                                                                                                                                                                                                                                                                                                                                                |
| <b>SIGN=0</b>         | After running lmcrypt, this will hold the signature associated with the INCREMENT line.<br><br>INCREMENT line with <i>licenseUser</i> is used to store information on the user to whom the license was issued.                                                                                                                                                                                                                                                                                                       |
| <i>PAK</i>            | Product Authorization Key.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|                       | Additional INCREMENT lines are for devices. Each additional device type has an INCREMENT line corresponding to it.                                                                                                                                                                                                                                                                                                                                                                                                   |
| <i>ExpirationDate</i> | Usually the same as the expiration date for the whole license disk.                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <i>DeviceLimit</i>    | If unlimited, -1 is used.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| #                     | Comment lines are started with this symbol.                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |

**CISCO CONFIDENTIAL****License File Format for Common Services 3.0**

Common Services 3.0 uses a new license file format.

Sample license file for an evaluation license for CDOne 3.0:

```
INCREMENT cdone cisco 3.0 31-dec-2004 uncounted \

VENDOR_STRING=<LicType>Evaluation</LicType><Code>ZZ</Code><Count>10</Count><CountType>Absolute</CountType><XINFO>LMS30</XINFO> \
 HOSTID=ANY \
 NOTICE="<LicFileID>lms</LicFileID><LicLineID>0</LicLineID> \
 <PAK>dummyPak</PAK>" SIGN=48CCDF82DDF6
```

Each application in the bundle is represented by an INCREMENT line that specifies the name and version of the application. The first line in the file describes the bundle itself and does not contain any information that is useful to the applications. In addition to the feature name and version, Common Services licensing uses the fields expiry date (set to 31-dec-2003 in this example) and VENDOR\_STRING to store the information it requires.

| Field                                | Description                                                                                                                                                                         |
|--------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>expiry date</b>                   | The specified expiry date for the evaluation licenses is not currently used but may be used in future to determine a date beyond which the evaluation license may not be installed. |
| Date                                 | For Evaluation License File.                                                                                                                                                        |
| <i>permanent</i>                     | For all other files.                                                                                                                                                                |
| <b>VENDOR_STRING</b>                 | This field can be used by a vendor to customize the license file.                                                                                                                   |
| License type(using the tag LicType)  | The value for this tag can be one of Evaluation, SEVT, NFR or NA.<br><br>Permanent licenses and upgrade licenses need not encode this information.                                  |
| Resource count (using the tag Count) | The number of devices (or any other resource) that the application is licensed for.<br><br>This value is -1 when no constraints apply.                                              |

**CISCO CONFIDENTIAL**

| Field                                                      | Description                                                                                                                                                                                                                                                                                                                                                                        |
|------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Interpretation of resource count (using the tag CountType) | <p>Indicates how the value specified for resource count is to be interpreted.</p> <p>Values for this tag are:</p> <ul style="list-style-type: none"> <li>• Cumulative— the specified count is to be treated as an addition to any prior license for combination of application and version.</li> <li>• Absolute— the specified count should override any prior license.</li> </ul> |
| Application code (using the tag Code)                      | <p>The code assigned to this combination of application and version by DNMBU marketing.</p> <p>The licensing component requires this tag for deciphering PINs. This tag is present only in evaluation license.</p>                                                                                                                                                                 |

**Alternate License File Format**

License files issued by Swift will employ an alternate license file format to represent license information. This format uses other FLEXlm license file fields as follows to represent the information stored in the VENDOR\_STRING field in the format described in the previous section.

Sample license file for the alternate format:

```
SERVER
VENDOR cisco
UPGRADE CDONE cisco 2.2 3.0 permanent 10 \
 VENDOR_STRING=<XINFO>LMS30</XINFO> \
 NOTICE="<LicFileID>lms</LicFileID><LicLineID>0</LicLineID> \
 <PAK>dummyPak</PAK>" SIGN=535525A647B6
FEATURE RME cisco 4.0 permanent 1500 \
 VENDOR_STRING=<XINFO>LMS30</XINFO> \
 NOTICE="<LicFileID>lms</LicFileID><LicLineID>0</LicLineID> \
 <PAK>dummyPak</PAK>" SIGN=7078C85E40E8
FEATURE DFM cisco 2.0 permanent 1000 \
 VENDOR_STRING=<XINFO>LMS30</XINFO> \
 NOTICE="<LicFileID>lms</LicFileID><LicLineID>2</LicLineID> \
 <PAK>dummyPak</PAK>" SIGN=52A22CF20DF4
FEATURE CampusManager cisco 4.0 permanent 500 \
 VENDOR_STRING=<XINFO>LMS30</XINFO> \
 NOTICE="<LicFileID>lms</LicFileID><LicLineID>3</LicLineID> \
 <PAK>dummyPak</PAK>" SIGN=333AD18EF768
```



**CISCO CONFIDENTIAL**

| Field            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| License type     | License files issued by Swift represent either Permanent or Upgrade licenses. UPGRADE lines represent upgrade licenses while FEATURE and INCREMENT lines represent permanent licenses.<br><br>An upgrade license results in the device/resource counts from previous versions to be carried over. When an upgrade license is used to increase the resource count, an additional INCREMENT or FEATURE line will be present to indicate the increased count. |
| Resource         | The count will be encoded in the license count field of FLEXlm license files. The value of this field is uncounted when no restrictions apply.                                                                                                                                                                                                                                                                                                             |
| Count type       | Resource counts in INCREMENT lines are Cumulative while the counts in FEATURE lines are Absolute.                                                                                                                                                                                                                                                                                                                                                          |
| Application code | Permanent and Upgrade licenses will not contain an application code.                                                                                                                                                                                                                                                                                                                                                                                       |

**Proof-of-Purchase (POP)**

Proof-of-Purchase (POP) will be obtained when you enter an upgrade license key. The customer will not be able to install a product if they have paid only for an upgrade license, unless the customer has got a the license for a previous version.

One way of ensuring that a customer has purchased a previous version of the product involves checking for the previous version installation CD. The customer will only be required to use a CD as a last resort, even in a case where the upgrade is done on a new machine and the previous version of the product is one another machine. If a CD check is ultimately needed, the licensing component must *not* require the user to remove a CD that is being used to install the upgrade, replace it with a CD for the previous version to do the POP, then reinsert the product upgrade CD.

The licensing component will perform a POP check by checking the license file of the previously installed version of a definitive product for which the upgrade license has been purchased.

Versions of applications that are not based on CS 3.0 such as RME 3.4, RME 3.5, DFM 1.1, DFM 1.2, Campus Manager 3.2, and Campus Manager 3.3 have license files to help implement the CMF-style licensing. These files are called rme.xml, dfm.xml and cm.xml respectively for RME, DFM and Campus Manager respectively. These files contain an EXPIRY line with two possible values: NEVER (for purchased license) and 90 (for an evaluation license).

One application per bundle will be used to identify which bundle the customer had purchased. The following table shows what needs to be checked per bundle:

**CISCO CONFIDENTIAL****Table 34-5 Files to be Verified by Applications**

| Bundle/Licensing Component Consumer | Definitive Application   | Files               | Comments                 |
|-------------------------------------|--------------------------|---------------------|--------------------------|
| LMS                                 | RME                      | rme.xml             |                          |
| RWAN                                | ACLM                     | N/A                 | Expected to be EOS'd.    |
| ITEM                                | ITM                      | itm.xml             |                          |
| CVM                                 | CVM                      | cvm.xml             |                          |
| EMS                                 | RME                      | rme.xml             |                          |
| Cable Mgr                           | RME                      | rme.xml             |                          |
| QPM                                 | QPM                      | qpm.xml             |                          |
| SNMS                                | RME                      | rme.xml             |                          |
| VMS                                 | Core                     | N/A                 | Use license in database. |
| ACLM/IPM add-on to LMS              | ACLM or IPM respectively | aclm.xml or ipm.xml |                          |

**In-Place Upgrade**

Install can automatically check for Proof-of-Purchase if the customer enters an upgrade license key or PIN, the proof of whether the customer has a purchased previous version is available immediately from the installed version.

**Remote Upgrade**

Install must prompt for Proof-of-Purchase. If the customer enters an upgrade license key or PIN, the proof of whether the customer has a purchased previous version must be verified.

- **Upgrade License Key is entered at install time**

At the end of installation, a message similar to the one below is displayed:

```
You have entered an upgrade license key. Please run the program
<NMSROOT>/bin/validate_upgrade.exe to validate that this is an upgrade.
```

Until the user runs the above program, the product will run in NAG mode.

- **Upgrade PIN entered at install time**

At the end of install, a message similar to the one below will be displayed:

```
"You have entered an PIN and not a license key. The product will continue to work in
nag mode from the date of installation. However, please obtain a valid license key
from CCO in order to make the product fully functional.
```

```
After obtaining the license key, load it into CiscoWorks by clicking on: Common
Services > Server > Admin > Licensing.
```

```
After entering the license key, please run the program <NMSROOT>/bin/
validate_upgrade.exe to validate the upgrade."
```

Until the user runs the above program, the product will run in NAG mode.

- **Upgrade License Key entered through GUI After the install**

**CISCO CONFIDENTIAL**

After entering the license through the GUI, a message will be printed similar to the following:

```
You have entered an upgrade license key. Please run the program
<NMSROOT>/bin/validate_upgrade.exe to validate the upgrade.
```

Until the user runs the above program, the product will run in NAG mode.

**License CLI****Note**

This section is applicable only to previous VMS based license.

The following programs are provided to support the installation of licenses:

**Table 34-6 License Program CLIs**

| License CLI     | Description                                                                                                                            |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------|
| validatelicense | Verifies whether the supplied license is valid and whether it can be supported by CMF Licensing<br><b>Location:</b><br>NMSroot/MDC/bin |
| addlicense      | Upgrade customers can use this utility if the upgrade check fails during the install.<br><b>Location:</b><br>NMSroot/bin               |

***CISCO CONFIDENTIAL***