Hexagon Safety & Infrastructure/Hexagon Geospatial

Concurrent License Details for Hexagon Safety & Infrastructure/Hexagon Geospatial Products

Intergraph Corporation has elected to do business as: “Hexagon Safety & Infrastructure,” in certain public safety, utility, communications, transportation, and information technology markets; “Hexagon Geospatial,” in certain geospatial markets; and, “Process, Power & Marine,” in certain engineering markets. Use of these alias and trade names does not reflect any change of legal corporate entity, applicable tax identification number, or similar formalities.

This document has been updated at the highest level to reflect the new name of Intergraph Security, Government & Infrastructure.

Concurrent License Details for Hexagon Safety & Infrastructure/Hexagon Geospatial Products

September, 2015

Table of Contents

1. Introduction

Many of the software applications developed by Hexagon Safety & Infrastructure/Hexagon Geospatial division have the capability to use the FlexNet Publisher (FNP) concurrent licensing option. Concurrent licensing, also known as floating licensing, allows software licenses to be used anywhere on the network instead of being tied to specific computer hardware (as with nodelocked licenses). For example, you may have twenty people in your department who need to access a specific application, but, at any given time, only five copies may be in use. Concurrent licensing gives your users the ability to use the application as needed without requiring extra licenses. As each workstation initializes the application, a license is checked out from the license server. When all licenses are checked out, no one else can access the application until one of the current users is finished. Using concurrent licensing allows a larger group of people to make use of a limited pool of licenses. Most of the Hexagon Safety & Infrastructure/Hexagon Geospatial concurrent licenses can also be borrowed from the license server when the client system needs to go off the network. This is essential when a user needs to run the application on a laptop while away from the office. A borrowed license acts like a nodelock license when the laptop is disconnected from the network.

FlexNet Publisher based concurrent licensing uses the following basic components:

- **LMTools** – A windows application containing a variety of utilities that manage the license manager server and its settings. LMTOOLS also contains some client side utilities that can be used for license borrowing or in remotely determining the status of the license server.

- **lmgrd.exe** - This is the license manager daemon. It controls the license traffic for the application vendor daemons. It is required for concurrent licensing to run and is used on the license server.

- **Vendor daemon(s)**

  - **INGR.exe** – This is the vendor daemon used by all Hexagon Safety & Infrastructure/Hexagon Geospatial applications and is required for all products released prior to 2012. The vendor daemon is accessed by the license manager daemon.

  - **INGRTS**.exe – This is the required vendor daemon used by most Hexagon Safety & Infrastructure/Hexagon Geospatial applications released in 2012 or later. Note that both vendor daemons can reside on the same machine as the license manager service, but each requires a separate instance of lmgrd.exe.

- **The license pool file** – This is a text file with a .lic extension. It contains all the concurrent application licenses that are used with the specific vendor daemon.

- **The client application** – The specific Hexagon Safety & Infrastructure/Hexagon Geospatial application you are using on the client system..

LMTools integrates several FlexNet Publisher utilities into one interface. It's primarily used to configure concurrent licensing as a service on the system chosen to be the license manager server. This system does not have to be a robust machine as the license manager software does not utilize much CPU or memory. Within the license manager service, the license manager daemon (lmgrd.exe) manages the contact between the client application program and the appropriate vendor daemon. The vendor daemon keeps track of how many licenses are checked out and who has them. If the vendor daemon terminates for any reason, all users lose their licenses (though this does not mean the applications suddenly stop running because the applications will continue to run until exited.) Users normally regain their licenses automatically when the license service restarts the vendor daemon.

On a client system, **LMTools** can be used to initiate license borrowing. If the application license allows for borrowing, then any application client using the license can borrow a license. The license manager server can control whether licensing is allowed via its options file but it is the client that both initiates borrowing and facilitates the early return of the license. See section 3.2 for more information.

Licensing data is stored in a text file called the license pool file. The license pool file is created and installed by the license administrator. It contains information about the server nodes and vendor daemons and has at least one entry, called the INCREMENT line, for each licensed product. Each INCREMENT line contains a license key based on the Host ID specified in the SERVER line, the total number of licenses available, and other vendor-specific data. The increment section contains the actual licenses that are generated from the licensing web site. Additional licenses can be added to the pool file at any time; however, adding a new license will require re-initializing the license service by either performing a re-read of the license file from within LMTOOLs or restarting the license service.

Very little CPU time is required to run a license server, especially when the number of clients is small. For large numbers of clients or for a high volume of license check-out/check-in activity, CPU time might become significant. Memory use is also small, and the combined license manager and vendor daemon processes typically uses about 3 MB of memory.

All network communication between a client application and a given vendor daemon is handled via TCP/IP. The client application and the license server can run on separate nodes on your local area network or across any wide-area network as long as a valid TCP/IP connection can be made between the client and the license server. While a license is checked out, a regular "heartbeat" is sent from the client to the server telling the server that the license is still in use. The heartbeat and the check-out/check-in processes typically send less than 1 K of data. This means the licensing should have no communication issues even with relatively slow networks. For Hexagon Safety & Infrastructure/Hexagon Geospatial applications, this heartbeat occurs every two minutes. If the heartbeat is not detected, the license is considered to be inactive. Inactive licenses can be automatically reclaimed by using the TIMEOUT parameter in the OPTIONS file.

If you are dealing with a very large number of licensed applications, with each application exchanging heartbeat messages with the same server, the network bandwidth used may become significant. If this is the case, you may want to split the license file between two or more different servers. You will place x number of licenses on server A and y number of licenses on server B where x+y=total purchased licenses. This is one type of license file redundancy. This type of license file redundancy can also provide backup licenses if the main license server fails for any reason as can multi-server redundancy but the number of licenses available for checkout is reduced by the number licensed on the failed license server. Three-server redundancy is also supported by Hexagon Safety & Infrastructure/Hexagon Geospatial.

Three server redundancy provides failover protection provided the network is running. With three server redundancy all three license servers operate to form a triad. The license servers send messages to each other to ensure at least two servers are running and communicating. A quorum is formed when at least two of the three are running and communicating with each other. If there is no quorum (two of the three machines available), licenses cannot be checked out.

When the user generates the three server redundant license file they must have three separate machines designated as the license manager servers. There will be a Primary, a Secondary, and a Tertiary. Primary is the first address designated when generating the license file, Secondary is the second and Tertiary is the third. Tertiary can never be the Master, only Primary or Secondary can be Master and responsible for delivering licenses. The requirements for three server redundancy are:

- All three servers must be on the same operating system
- The port used by lmgrd must be specified and be the same for all three servers.
- You must have reliable network communication.
- All three servers must reside on the same network subnet.

If you have products that were released prior to 2012, the Hexagon Safety & Infrastructure/Hexagon Geospatial vendor daemon, INGR.EXE is older than version 11.7 which means you cannot create the license file using keywords PRIMARY_IS_MASTER and HEARTBEAT_INTERVAL in the License file. When using the vendor daemon INGRTS.EXE, then PRIMARY_IS_MASTER can be used with three server redundant licenses.

Constant network communication between the three systems involved in redundancy is required because they exchange heartbeats periodically and poor communications can cause poor performance and even cause a failure to serve licenses If this is a concern, you may want to consider using a virtual system as a backup to your primary license server. Having a virtual backup of your physical license server allows you to quickly bring the virtual license server online in an event of a failure on the physical license server. You can also use a virtual system as your primary license server. By adopting the VM scenario, you eliminate the network congestion that may occur with the three server redundant servers.

2. Configuring the License Server

2.1 Generating a License Pool File

Hexagon Safety & Infrastructure/Hexagon Geospatial products use two different addresses for the concurrent license file. Products released prior to 2012 use the static TCP/IP address of the license server machine and the products released in 2012 use either the static TCP/IP address or a composite Host ID (displayed when you run the utility to identify the Host ID). To reiterate, the TCP/IP address of the license server must be static. It cannot be dynamically assigned (via DHCP). This is required because the TCP/IP address is included in the license key and is part of the license verification process. The composite Host ID is a Host ID generated from various components on the license server service machine. The application will display the appropriate composite Host ID for your system.

To generate your licenses, have both your server's TCP/IP address or Composite Host ID (if Composite is required, the License File Not Found message will display the required composite Host ID) and your License Authentication Code (LAC) available, and refer to the Hexagon Safety & Infrastructure/Hexagon Geospatial licensing web site at:

http://licensing.hexagonsafetyinfrastructure.com/flexnet/operationsportal

Detailed instructions on generating licenses are available online at both the License Generation web site, http://licensing.hexagonsafetyinfrastructure.com/flexnet/operationsportal or the License Resources website: https://support.hexagonsafetyinfrastructure.com/infocenter/index?page=licensing_resources

Here are the basic steps:

1. Go to http://licensing.hexagonsafetyinfrastructure.com/flexnet/operationsportal

2. Key in your License Authentication Code (LAC) at the prompt and click **Login**

    a. Click **Generate License** to show the list of products.

    b. If you are licensing multiple applications for the same license pool file, you may select all of them to generate the license file by clicking the check box by each product.

    c. Click Generate.

    d. If you are generating licenses for multiple products, select **"Yes"** for **Consolidate** generated licenses.

    e. You have the option of populating an email address to email the license file, but this is not a required field. Hexagon Safety & Infrastructure/Hexagon Geospatial recommends you save the license file generated rather than emailing, though emailing the license is an option you can complete later in the process.

f. Click Next.

g. Choose **Add New Host.**

   i. Select INTERNET as the Host ID type if you are using the static TCP/IP address.

      1. Enter the static TCP/IP address in this format;

      xxx.xxx.xxx.xxx

   ii. Select VENDOR DEFINED as the Host ID type if you are using the composite Host ID displayed by the product.

      1. Enter the Composite Host ID address in this format:

      COMPOSITE=xxxxxxxxxxxx

      (if there are multiple composite Host IDs, enter in this format:

      COMPOSITE=xxxxxxxxxxxx COMPOSITE=xxxxxxxxxxxx

h. Once you have entered the Host ID,
click OK.

i. Click **Next** to enter the quantity of licenses you are generating for each selected product.

j. Click **Verify** to ensure the license is valid.

k. Click **Generate** to generate the concurrent licenses and consolidate into a single license file.

l. The License Summary page will display giving you the option to **Save to File** or **Email License** and then **Complete**. Please do not click **Complete** without first using the **Save to File** or **Email License** buttons.

3. Create a directory on the system you have chosen as a license server. The name and location does not matter.

   You should download the latest LMGRD.exe and LMTOOLS.exe from https://licensing.hexagonsafetyinfrastructure.com/flexnet/operationsportal/. This is particularly important if you are using a 64-bit Windows 2008 server as your License Manager machine. LMGRD.exe and LMTOOLs.exe version 11.4 or higher is required for a 64-bit Windows 2008 server. The latest version of LMGRD.exe and LMTOOLs.exe is backwardly compatible with all license files and we recommend downloading the latest version. Do **not** download **lmadmin** as Hexagon Safety & Infrastructure/Hexagon Geospatial does not yet use this functionality.

Located on the product application CD is a top-level directory called Licensing. Copy the following files to the license directory you created on the license server:

a. INGR.exe

b. INGRTS.exe

c. Example_Ingr.opt

4. Copy the license file generated in Step 2.. to the folder created in Step 3.

5. You are now ready to create the license service.

2.2 Creating the License Service

To configure a license server, you must have Administrator privileges on the system that will be used as the license server. By default, the port number used by the license manager daemon will be automatically assigned from the range 27000-27009. The port number for the vendor daemon is used internally and is also automatically assigned. You can override the automatic port assignments by specifying the port number in the SERVER/VENDOR lines of the license file. To specify a port outside the default range, use the following example for a static TCP/IP address (in this case, the port is 40000 for lmgrd and 40001 for INGR):

```
SERVER 129.139.149.259 INTERNET=129.139.149.256 40000
VENDOR INGR PORT=40001

SERVER this_host COMPOSITE=5C1C1725722040000
VENDOR INGRTS PORT=40001
```

Note: The use of the syntax *PORT=* is not required on the SERVER line but is required on the VENDOR line. If you do change the port number, that information must also be used on the client. See Section 3 for more information.

The procedure for creating a license service on the license server is listed below:

1. Configure the license pool file.

   The license pool file is a text file using a .lic extension. You can name the license pool file anything you want but it must have a .lic extension. The contents of the file depend on the number of licenses you have and the number of applications being using the license service. The contents of a typical license pool file containing 35 licenses for one application, in this case GeoMedia_Professional, using TCP/IP and the INGR vendor daemon, is shown below:

```
SERVER this_host INTERNET=129.139.149.259
VENDOR INGR
USE_SERVER
#
INCREMENT GeoMedia_Professional INGR 6.1 31-mar-2013 35 \
        ISSUER="Hexagon" \ISSUED=26-jan-2012 BORROW=2160
        NOTICE="Hexagon FNO \Admin" SN=SF-1831-9613-2208-9219 START=25-
        jan-2012 SIGN="2CBB \BD4C F21B 8BB2 C82A 9007 455A C1C5 E633 CCA9
        3F78 DB66 C31F \0D22 48EB 077A 5911 29A2 1BDD 5A54 E42C 4B2C
        3F82 EEBC 2FF3 \B15F 9E74 3218 0DF9"
```

Make sure you change the notation *this_host* in the SERVER line to either the TCP/IP address or the system's DNS name. If you have multiple TCP/IP addresseson the server use the DNS name.

The contents of a typical license pool file containing 20 licenses for one application, in this case GeoMedia_Professional, using a COMPOSITE Host ID , the INGRTS vendor daemon, and configured for a server named LICSERVER1 is shown below:

```
SERVER LICSERVER1 COMPOSITE=5C1C17257220
VENDOR INGRTS
USE_SERVER

INCREMENT GeoMedia_Professional INGRTS 7.0 31-mar-2013 20 \
        ISSUER="Hexagon" \ISSUED=16-feb-2012 BORROW=2160
        NOTICE="Hexagon FNO \Admin" SN=SF-1831-9613-2208-9219 START=25-
        jan-2012 SIGN="1030 \FC6D B75B 700A B95B 9007 B47D C1D9 E633 CCA9
        3F78 DB66 C31F \DA71 48EB 077A 5911 29A2 1BDD 5A54 D804 8F8D
        5019 C351 4CBD \B4E3 D5F8 6C02 BCA5 4C78"
```

If multiple Hexagon Safety & Infrastructure/Hexagon Geospatial applications are licensed concurrently or additional licenses are added for an existing application, you need to add their entries to the license pool file. Use a text editor like Notepad, and copy the entire INCREMENT section from each separate license file into one large master pool file, as shown in the example below (# is used to mark a comment line and is optional):

```
SERVER LICSERVER1 INTERNET=129.139.149.259
VENDOR INGR
USE_SERVER
#
INCREMENT GeoMedia INGR 6.1 31-mar-2013 10 27F1F0914F08 BORROW=2160 \
        ISSUER="Hexagon" \ISSUED=5-mar-2012 NOTICE="This license is for
        internal use \only. It is assigned to Hexagon. It is not to be used for \
        production or placed on a customer's machine." \SN=SF-8371-1641-6631-
        5218 START=1-mar-2012 SIGN="024D BD1E \055B BF56 E8E7 B99A EFD4
```

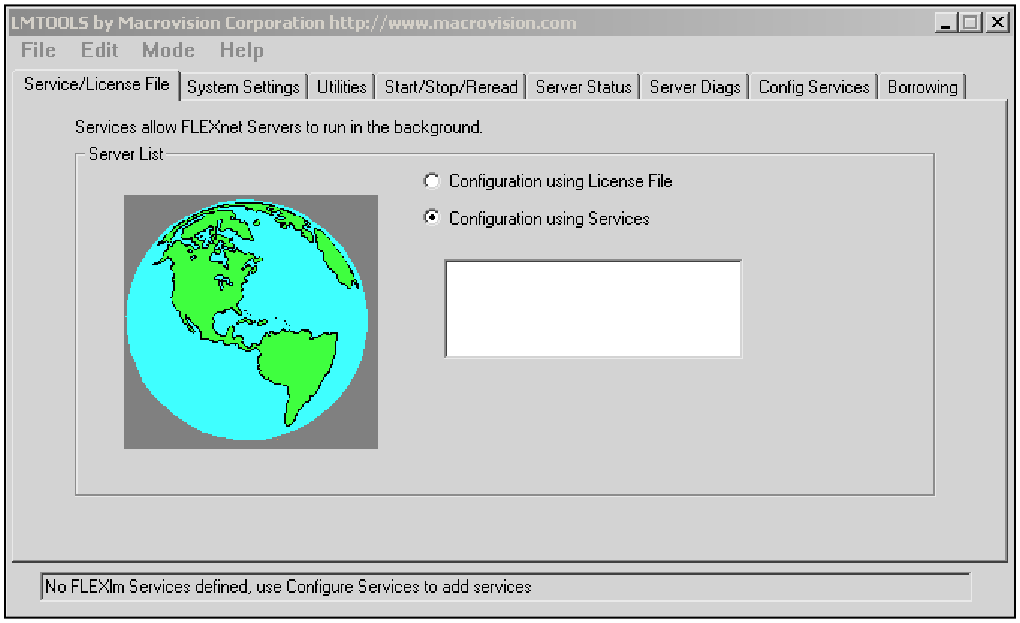
```
            499D 34A0 C0F7 9BB6 AF52 2A7F 359D \2E2C 1DDD 057F E03D D8CF
            68D2 309A F6C3 A544 8D02 E7B5 2352
#

INCREMENT ISSD INGR 6.2 31-mar-2013 30 E6541DB580C0 BORROW=2160 \
            ISSUER="Hexagon" \ISSUED=28-feb-2012 NOTICE="This license is for
            internal use \only. It is assigned to Hexagon. It is not to be used for \
            production or placed on a customer's machine." \SN=SF-1601-2013-1341-
            3602 START=1-mar-2012 SIGN="07AD 05D9 \D747 2887 C98E C1DB 8EEC
            A418 9035 43A1 EF3C C29E F278 1575 \AD48 1092 B7AB FA2F 4327
            DD3A 72C2 DE6C DF18 01BF 4B07 D034
#

INCREMENT GAdministrator INGR 10.0 31-mar-2013 10 765E8A05619B \
            BORROW=2160 ISSUER="Hexagon" ISSUED=5-mar-2012 NOTICE="This
            license is for \internal use only. It is assigned to Hexagon. It is not to \
            be used for production or placed on a customer's machine." \SN=SF-
            1311-6617-6126-2225 START=1-mar-2012 SIGN="05B5 7265 \5DE6 2AD0
            8951 3DDC 020C ECC3 1102 6393 924E 6897 2FA4 12B7 \4168 0A48 E6A0
            9AAF 99DB 444C DBAC C651 3D6B 9863 B39F C744 \45D5 EC53 61D9
            E80B"
```

Note: While the top 3 lines of the license file are editable, never edit the contents of the INCREMENT section. Doing so will invalidate the license. Always use a simple text editor like NotePad. Using a rich text editor like Wordpad may also invalidate the license file. These files contain a checksum and will interpret any edits as tampering. You will have to regenerate the license if the section is edited in any way. Fortunately, you can retrieve the original license from the license web site.

2. After you have set up the license pool file, you need to configure the license pool service. This is done using LMTOOLS. The LMTOOLS application is shown below:



Services allow FLEXnet Servers to run in the background.

○ Configuration using License File

● Configuration using Services

No FLEXlm Services defined, use Configure Services to add services

3. On the main LMTOOLS dialog box, select the **Configuration using Services** radio button under the **Service/License** File tab. This will set up the license pooling to run as a service.

   a. Select the **Config Services** tab. This is where you configure the license pool service. The following will step you through the process:

   b. Enter a service name, for example, *SGI Main License Server.*

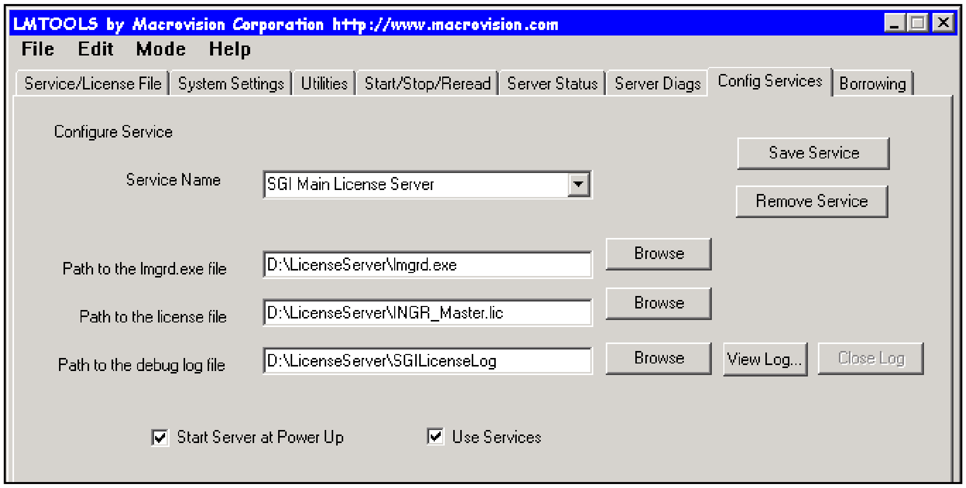
   c. Browse to select the license daemon file: *lmgrd.exe.*

   d. Browse to select the license pool file, for example*, LicensePool.lic.*

   e. Browse to select a location for the log file, and give the log file a name, for example, *LicenseLog.log.*

   f. Select the mode you wish to use. For best results, *start the server at power up*, and use *Windows services* by selecting the indicated boxes. These are optional settings, but they give the best results.

   g. Click **Save Service** to save the settings. From now on, when this server is booted, this license server will start automatically as a Windows service.


LMTOOLS by Macrovision Corporation http://www.macrovision.com
File Edit Mode Help
Service/License File | System Settings | Utilities | Start/Stop/Reread | Server Status | Server Diags | Config Services | Borrowing
Save Service
Remove Service
Browse
Browse
Browse
View Log...
Close Log

Configure Service

Service Name: SGI Main License Server

Path to the lmgrd.exe file: D:\LicenseServer\lmgrd.exe

Path to the license file: D:\LicenseServer\INGR_Master.lic

Path to the debug log file: D:\LicenseServer\SGILicenseLog

☑ Start Server at Power Up ☑ Use Services

4. To immediately start the service, select the **Start/Stop/Reread** tab, and then click **Start Server**.To check the availability of licenses and to see if there are any problems with the license file, check the **Server Diags** tab, or **click View Log** on the **Config Services** tab.

   a. On the **Server Diags** tab, select **Perform Diagnostics**. The main field on the dialog box will list the license status for each license in the pool file.

   b. On the **Config Services** tab, clicking **View Log** will list the contents of the license log file. This file will tell which licenses are checked out and who the license users are. It will also indicate how many licenses are still available.

Note: You cannot use remote mounted disks for any of the components involved in licensing. All the components must reside locally on the system hosting the license service.

2.3 Adding New Licenses to the Pool File

Additional licenses can be added to the license pool file a any time. You can add additional numbers of licenses for existing licensed applications by simply adding a new increment section containing more licenses. Similarly, as you license new applications, you can add their increment sections to the license pool file. For the license server to recognize changes to the license file, go to the **Start/Stop/Reread** tab in **LMTOOLS** and select the **ReRead License File** button.

2.4 Compatibility Issues

FlexNet Publisher is used by a wide variety of software vendors. Each vendor may be on a different version of the FlexNet Publisher software and may have different capabilities.

If you are running multiple applications from the same vendor, put the license entries (INCREMENT sections) for each of the applications into the same global license pool file. If you have multiple vendors utilizing concurrent licensing, you may need to take steps to prevent licensing conflicts during installation. There are several ways to accomplish this:

- Use multiple license server nodes, each running one *lmgrd* and one license file. This means you will set up multiple machines as your license servers. The advantage is the licensing load is distributed and each server has its own debug log. The main disadvantage is high administrative overhead costs.

- Install one license server node running one *lmgrd* and several license files. Using this technique, the one lmgrd process serves two or more vendor daemons, each with one or more license files and all license files reside in the same location. The advantage is lower administrative overhead, but if lmgrd goes down, all licensing is disabled and all output from all vendor daemons goes into a single debug log which makes it harder to debug a specific vendor daemon’s problem.

- Install one license server node running multiple *lmgrds* and multiple license files. The advantages of this technique are the license files remain separate and each has its’ own lmgrd so if one service goes down, only the licenses served by that instance of lmgrd are not available. Another advantage is a separate debug log file for each vendor daemon. However the administrative overhead for this option is high and if the machine goes down all licensing is disabled, plus the licensing load is concentrated on one machine.

License files for two different FlexNet Publisher-licensed products may or may not be using the same version of FlexNet Publisher. FlexNet Publisher should handle this situation as long as the following guidelines are used:

- A newer *lmgrd* can be used with an older vendor daemon, but a newer vendor daemon might not work properly with an older *lmgrd*.

- A newer vendor daemon (or *lmgrd*) can be used with an older client application, but a newer client application might not work properly with an older vendor daemon.

In order for you to have the broadest compatibility, you need to ensure that the following rules are adhered to:

The version of LMTOOLS used must be greater than or equal to

- the version of *lmgrd*, which must be greater than or equal to
- the version of the vendor daemon (in our case, *ingr*), which must be greater than or equal to
- the version of the FlexNet Publisher-licensed application, which must be greater than or equal to
- the version of the license file format used.

2.5 Redundant Licensing

The system acting as license server must be stable and not subject to frequent reboots and shutdowns. As long as the server remains up, the licenses are available for use. If, for any reason, the license server fails, the client systems may be unable to retrieve a license for use. If you have a mission-critical environment and you serve licenses to a large number of users, you may want to set up more than one license server. FlexNet Publisher supports the following two methods of license redundancy:

- Three Server Redundancy
- License File Redundancy

With *three server redundancy*, you need a special license file that is configured for three different servers. The three license servers communicate with each other, forming a quorum. As long as any two servers remain in the quorum, licenses are served to the clients. This configuration provides failover, but it does not provide load balancing. When you configure your client for 3 server redundancy, set the INGR_LICENSE_FILE variable as follows:

```
INGR_LICENSE_FILE=port@primaryserver,port@secondaryserver, port@tertiaryserver
```

In this case, the use of the port is required and each redundant server must be listed (primary first) separated by commas (,).

With *license file redundancy*, each license server serves as a subset of the total number of licenses available. This client is then configured to access each server, in order, using the INGR_LICENSE_FILE variable (more on this in section 3For example, if you have purchased 20 licenses, you should put 10 on one server and 10 on another. When you access your LAC, you can specify the internet address and the number of licenses you want on each server. When you configure your client, set the INGR_LICENSE_FILE variable as follows:

```
INGR_LICENSE_FILE=@server1;@server2
```

You can configure as many servers as you want using this method.

A third alternative supported by Hexagon Safety & Infrastructure/Hexagon Geospatial is to configure a virtual server identical to your license server, leave it turned off (necessary because of network issues) and turn it on in the event that your license server fails.

2.6 Licensing Options

When managing your concurrent licenses, you can use the default mode, or you can select from a number of options that control how the licenses are used. A license administrator can use the options file to further secure or to open the use of the licenses available on the server. Usage is controlled by user login, by machine address, or by groups. Specifically, the license administrator can do the following:

- Allow the use of a license
- Deny the use of a license
- Reserve licenses for specific users
- Control the amount of information logged about license usage

2.6.1 Options File

The following are required in order to use an options file with your concurrent license:

1. Name: By default, the options file needs to be the same name as the vendor daemon. In this case, the vendor daemon is *INGR*, so the options file should be called *INGR.opt*. You can use a different name if you use the options parameter and specify the full path to the options file.

2. Location: The easiest way to use the options file is to include it in the same directory as the vendor daemon INGR.exe. You can override this by adding the option file path and/or name to the VENDOR line of the license pool file, as shown in the example below:

   ```
   # Default Method:
   VENDOR INGR
   # Using a different options file name:
   VENDOR INGR options=INGROptions.OPT
   # Using a different options file path:
   VENDOR INGR options=c:\path\INGR.OPT
   ```

3. If the license service is already running, you will need to restart it in order to use the options file. This can be done using **LMTOOLS > Start/Stop/ReRead > Stop Server** followed by **Start Server.**

Note: By default, the vendor daemon (INGR.exe in this case) will always search for an options file with the name *vendor.opt* (*ingr.opt* in this case). If you follow this naming convention and place the options file in the same folder as the vendor daemon file, there is no need to edit the license file to add the “options=” parameter. If you do not want the options file to be used, you will have to rename it or remove it from the directory.

When the vendor daemon is started by *lmgrd*, the vendor daemon reads its options file. Only one options file is allowed per vendor daemon. If any changes are made in the options file, you must restart the license service so that the file can be re-read by the vendor daemon. You can use LMTOOLS for this.

2.6.2 Options Syntax

Each line in the options file is limited to 2048 characters. The “\” character is a continuation character in the lines of an options file. You can include comments in your options file by starting each comment line with a pound sign (#). Everything in an options file is case sensitive. Be sure that user names and feature names are entered correctly.

When editing an options file, the following terminology is used:

- **user** – The OS user name of the user executing the program (the domain does not need to be included in the user name).
- **host** – Computer system where the application is executing. This can be the DNS name or the actual TCP/IP address.
- **feature** – Name of the application being licensed (for example, GeoMedia).
- **type** – Can be any one of the following: USER, HOST, ADDRESS, GROUP, or HOST_GROUP.

A licensed feature (application) can be distinguished with an optional keyword. This is useful when you need to distinguish a particular group of licenses if there are multiple INCREMENT lines for a single feature. The following syntax is used:

```
feature:keyword=value
```

The available keywords for use with most Hexagon Safety & Infrastructure/Hexagon Geospatial application licenses include *VERSION*, *ISSUER*, and *NOTICE*.   For example:

```
GeoMedia_Professional:VERSION=6.1
```

Not all of the available FlexNet Publisher options can be used with Hexagon Safety & Infrastructure/Hexagon Geospatial application licenses, but most are available. An example options file, *Example_ingr.opt*, is available in the Licensing directory on the CD. For more information on using options, refer to the *FlexNet Publisher End Users Guide*.

The following options are supported with the INGR.exe vendor daemon:

BORROW_LOWWATER: Sets the number of licenses (n) that *cannot* be borrowed. This ensures that you will always have some licenses remaining on the license server.

Syntax:

```
BORROW_LOWWATER feature[:keyword=value] n
```

Example (the application in this case is GeoMedia Professional):

```
# Ensure that 5 licenses cannot be borrowed from the server.
BORROW_LOWWATER GeoMedia_Professional 5
```

EXCLUDE: EXCLUDE overrides INCLUDE. Excludes a list or pre-defined group of users, or hosts, in the list of who is allowed to use licenses for this feature. Anyone not in an EXCLUDE statement will be allowed to use that feature.

Syntax:

```
EXCLUDE feature type {list | group_name}
```

Examples (the application in this case is ImageStation Stereo Display):

```
# In all cases, the application in question is GeoMedia
# but you can substitute any licensed application.
# Prevent user1 from checking out a license.
EXCLUDE ISSD USER user1
# Prevent the host1 computer from checking out a license.
EXCLUDE ISSD HOST host1
# Prevent a group of computers from checking out a license
# either by a list or by a named HOST_GROUP H1.
EXCLUDE ISSD HOST_GROUP host1 host2 host3
EXCLUDE ISSD HOST_GROUP H1
# Prevent a group of users from checking out a license
# either by a list for by a named GROUP G1.
EXCLUDE ISSD GROUP user1 user2 user3
EXCLUDE ISSD GROUP G1
# Prevent a network address from checking out a license.
EXCLUDE ISSD INTERNET 123.234.456.678
EXCLUDE ISSD INTERNET 123.234.456.*
```

EXCLUDEALL: Excludes a list or pre-defined group of users, or hosts, in the list of who is allowed to use all features served by this vendor daemon. Anyone not in an EXCLUDEALL statement will be allowed to use these features.

Syntax:

```
EXCLUDEALL type {list | group_name}
```

Example:

```
# Similar to EXCLUDE except feature does not need to be
# listed.
EXCLUDEALL USER user1
```

EXCLUDE_BORROW: Denies a list or pre-defined group of users from being able to borrow a license.

Syntax:

```
EXCLUDE_BORROW type {list | group_name}
```

Example:

```
EXCLUDE_BORROW USER user1
```

GROUP: Defines a group of users for use in INCLUDE, INCLUDEALL, EXCLUDE, EXCLUDEALL, and RESERVE option lines. Multiple GROUP lines for the same group will add all the specified users into the same group.

Syntax:

```
GROUP group_name user_list
```

Example:

```
GROUP G1 user1 user2 user3
```

HOST_ GROUP: Defines a group of computer names for use in INCLUDE, INCLUDEALL, EXCLUDE, EXCLUDEALL, and RESERVE option lines. Multiple HOST_GROUP lines for the same host group will add all the specified users into the same host group. Anywhere a host name can be used in an options file, an IP address can be used instead (wild cards can be used in an IP address).

Syntax:

```
HOST_GROUP group_name host_list
```

Examples:

```
HOST_GROUP HG1 host1 host2 host3
HOST_GROUP HG2 123.456.*.* 122.223.32.*
```

INCLUDE: Includes a list or pre-defined group of users, or hosts, in the list of who is allowed to use licenses for this feature. Anyone not in an INCLUDE statement will not be allowed to use that feature.

Syntax:

```
INCLUDE feature type {list | group_name}
```

Examples (the feature in this case is GAdministrator):

```
# In all cases, the application in question is GAdministrator
# but you can substitute any licensed application.
# Allow user1 to check out a license.
INCLUDE GAdministrator USER user1
# Allow computer host1 to check out a license
INCLUDE GAdministrator HOST host1
# Allow a group of computers to check out a license
# either by a list or by a named HOST_GROUP H1.
INCLUDE GAdministrator HOST_GROUP host1 host2 host3
INCLUDE GAdministrator HOST_ROUP H1
# Allow a group of users to check out a license
# either by a list or by a named GROUP G1.
INCLUDE GAdministrator GROUP user1 user2 user3
INCLUDE GAdministrator GROUP G1
# Allow a network address to check out a license.
INCLUDE GAdministrator INTERNET 123.234.456.678
INCLUDE GAdministrator INTERNET 123.234.456.*
```

INCLUDEALL: Includes a list or pre-defined group of users, or hosts, in the list of who is allowed to use all features served by this vendor daemon. Anyone not in an INCLUDEALL statement will not be allowed to use these features.

Syntax:

```
INCLUDEALL type {list | group_name}
```

Example:

```
# Similar to INCLUDE except feature does not need to be
# listed.
INCLUDEALL USER user1
```

INCLUDE_BORROW: Allows a list or pre-defined group of users, or hosts, in the list to borrow a license.

Syntax:

```
INCLUDE_BORROW type {list | group_name}
```

Example:

```
INCLUDE_BORROW USER user1
```

MAX: Sets maximum number of licenses (n) that can be used by a group or user.

Syntax:

```
MAX n feature type {list | group_name}
```

Example (the application in this case is GeoMedia) :

```
MAX 2 GeoMedia GROUP G1
```

MAX_BORROW_HOURS: Changes the maximum borrow period for the specified feature.

Syntax:

```
MAX_BORROW_HOURS hours feature type {list | group_name}
```

Example (the application in this case is GeoMedia):

```
MAX_BORROW_HOURS 100 GeoMedia GROUP G1
```

NOLOG: Suppresses the logging of the selected type of event in the debug log file.

Syntax:

```
NOLOG {IN | OUT | DENIED | QUEUED}
```

Example:

```
NOLOG DENIED
```

RESERVE: Reserves a specific number of licenses (n) for a specific user or a group. Any licenses reserved for a user/group are dedicated to that user/group. Even when that user/group is not actively using the license, it will be unavailable to other users/groups.

Syntax:

```
RESERVE n feature type {list | group_name}
```

Example (the application in this case is ImageStation Stereo Display):

```
RESERVE 1 ISSD USER user1
```

TIMEOUT: Sets the time after which an inactive license is freed and reclaimed by the vendor daemon. If the server does not receive a regular heartbeat from the feature, the features license is considered to be inactive.

Syntax:

```
TIMEOUT feature[:keyword=value] seconds
```

Example (the application in this case is GTechnology):

```
TIMEOUT GTechnology 180
```

TIMEOUTALL: Sets the time after which an inactive license is freed and reclaimed by the vendor daemon for all licensed features.

Syntax:

```
TIMEOUTALL seconds
```

Example: Set the timeout to 5 minutes.   If a heartbeat is not received from the client in a five minute period, reclaim the license. This is useful if licenses are being held by an application that has crashed.

```
TIMEOUTALL 300
```

The TIMEOUTALL parameter is the most common entry in the options file. Never set the TIMEOUT to be less than the standard heartbeat which is 120 seconds. For best results, the minimum timeout should be 300 (5 minutes). This allows for a missed heartbeat due to a network interruption.

2.6.3 License Server Example with Options File

The following is a simple example of a license pool file that has been modified over time:

The license server is on the node 129.139.149.159.   Originally, there were five GeoMedia Professional 6.0 licenses available for use. The license administrator added ten new GeoMedia Professional 6.0 licenses and twenty GeoMedia 6.0 licenses. The management team has made the following requests:

- The Admin group needs to have two GeoMedia Professional 6.0 licenses available to them at all times.
- The Maintenance group needs three GeoMedia licenses available at all times.
- Two common systems always need to have a license available for GeoMedia and GeoMedia Professional.

For this example, here are the contents of the license file as it resides on the server:

```
SERVER 129.139.149.159 INTERNET=129.139.149.159
VENDOR INGR
USE_SERVER
```

```
#
INCREMENT GeoMedia_Professional INGR 6.1 permanent 5 BORROW=2160
2F60C6293318 ISSUER="Hexagon"…
INCREMENT GeoMedia_Professional INGR 6.1 permanent 10 BORROW=2160
03D0FF1783C2 ISSUER="Hexagon "…
INCREMENT GeoMedia INGR 6.1 permanent 20 6BD10168398F BORROW=2160
ISSUER="Hexagon"…
```

Here is an example of the contents of a typical *INGR.opt* options file for the above configuration. In this options file, the first line sets TIMEOUTALL to ensure that licenses are not locked when an application fails. The BORROW_LOWWATER parameter ensures that there will be at least five GeoMedia Professional licenses available on the server. MAX_BOROW_HOURS 168 ensures that no one can borrow a license for more than a week. The Maintenance group and the Admin groups both have three members. The usernames specified are unqualified network usernames. The host group is made up of the addresses from two different systems. In the RESERVE section, three GeoMedia Professional licenses are reserved for the Admin group, three GeoMedia licenses are reserved for the Maintenance group, and two GeoMedia licenses are reserved for the HG1 group.

```
TIMEOUTALL 300
BORROW_LOWATER GEOMEDIA_PROFESSIONAL 5
MAX_BOROW_HOURS 168 GEOMEDIA_PROFESSIONAL
#
GROUP ADMIN cewoodbu rsbandha rcbooth
GROUP MAINTAINENCE vadlani jlriddle bjbeckma
HOST_GROUP HG1 123.456.789.123 122.223.32.25
#
RESERVE 2 GeoMedia_Professional GROUP ADMIN
RESERVE 3 GeoMedia GROUP MAINTAINENCE
RESERVE 2 GeoMedia HOST_GROUP HG1
```

3. Configuring Client Systems

In past releases, Hexagon Safety & Infrastructure/Hexagon Geospatial applications had specific requirements regarding license files. This is no longer the case. Individual license files are no longer required, and the name of the client license file can be anything with a **.lic* extension. There is a default location for license files, but users can customize this to suit their needs. For concurrent license users, there is no requirement at all for a local file. The only requirement for the client is to indicate the location of the license server.

FlexNet Publisher provides the following two variables that can be used by the client to locate the license server:

- **LM_LICENSE_FILE** – Global license file location. This applies to all applications using FlexNet Publisher, regardless of the vendor. If you are using applications from only one vendor, you can use this variable. Otherwise, it may cause some applications to work and others to fail.

- **INGR_LICENSE_FILE** – INGR license file location. Any applications utilizing the INGR vendor daemon will look for their licenses in the locations specified here. Using this variable will ensure that you do not interfere with the operation of other FlexNet Publisher-based applications.

Either variable can be used; however, only INGR_LICENSE_FILE will be set up automatically during installation. Both variables can be set in the registry under the key *HKEY_LOCAL_MACHINE\SOFTWARE\FlexNet Publisher License Manager* or as local environment variables. The information in the following sections refers to INGR_LICENSE_FILE, but the same information applies if you are using LM_LICENSE_FILE.

3.1 Basic Configuration

Hexagon Safety & Infrastructure/Hexagon Geospatial applications look for licenses using the variable INGR_LICENSE_FILE, which can be set in the registry, as a local environment variable, or both. This variable may contain the path to the directory containing the license files, the full path to a specific license file, the DNS name (or TCP/IP address) of the license server, or any combination thereof (using a semicolon as a delimiter). The three most common ways to use this variable are listed below:

- In the registry under: HKEY_LOCAL_MACHINE\SOFTWARE\FlexNet Publisher License Manager. Set the variable INGR_LICENSE_FILE to the location of the client license file. This is the default and requires a client license file that points to the license server.

- In the registry under: HKEY_LOCAL_MACHINE\SOFTWARE\FlexNet Publisher License Manager. Set the variable INGR_LICENSE_FILE to the location of the server using the syntax @*servername*. This is the easiest method to use, and it negates the need for a local client file.

- Set the client environment variable INGR_LICENSE_FILE to either the location of the server or to the local client license file location. Use either a USER variable for a single-user access or a SYSTEM variable for all users on the system. This method is required if you want to use early return with license borrowing.

Note: Currently, the FlexNet Publisher early-return process forces the use of a local environment variable when using INGR_LICENSE_FILE. This is not required if you use LM_LICENSE_FILE.

During installation, the registry key HKEY_LOCAL_MACHINE\SOFTWARE\FlexNet Publisher License Manager is automatically created and set to the following location:

```
C:\Program Files\Common Files\Intergraph\Ingr_licenses
```

All license files should be put in this directory, and every user of the system will require read access to the files here. This is primarily for nodelocked licenses, but you can put a client-side concurrent license here as well. A client-side concurrent license is simply a subset of the license pool file on the server. For example, if the license pool file looks like the example below:

```
SERVER 129.139.149.219 INTERNET=129.139.149.219
VENDOR INGR
USE_SERVER
#
INCREMENT GeoMedia_Professional INGR 6.1 permanent 35 \ 0389FE05A7C6
ISSUER=”Hexagon”
#
INCREMENT ISSD INGR 6.2 permanent \ 25 9FA3BE6OECF9 ISSUER=”Hexagon”
```

Then the contents of the client file would be (remember, the filename must have a **.lic* extension) the following:

```
SERVER 129.139.149.219 INTERNET=129.139.149.219
VENDOR INGR
USE_SERVER
```

The same license file can be used on every client. You are not required to configure the license files in this manner. You can simply copy the entire license pool file to the client. The licensing software will ignore non-pertinent portions of the license file. If you have specified port numbers in the license pool file, make sure they are specified here as well.

The easiest way to configure the client is simply to point it at the license server. Normally the registry is used, and the following key determines the location of the license files:

```
HKEY_LOCAL_MACHINE\SOFTWARE\FlexNet Publisher License Manager
```

Under this key is the following registry variable:

```
INGR_LICENSE_FILE
```

By default, this value is set to:

```
C:\Program Files\Common Files\Intergraph\Ingr_Licenses
```

If you are using a server called *mylicserver*, you could specify the following in `INGR_LICENSE_FILE`:

```
@mylicserver
```

Or if you have configured a specific port: (for example, 40000):

```
40000@mylicserver
```

You could also replace mylicserver with the TCP/IP address of the server.

```
@123.456.789.256
```

Or with a port specified:

```
40000@123.456.789.256
```

Using this syntax tells FlexNet Publisher to look directly on the server for a license. If this syntax is used, no other information needs to be set on the client, and no specific files are required. This is the preferred way of configuring the client.

Other paths can be appended to this variable using a semicolon as a separator. For example, if you have a mix of concurrent and nodelocked licenses, you would want to specify a path to both, as shown below:

```
C:\Program Files\Common Files\Intergraph\Ingr_Licenses; @mylicserver;
```

The order of the path list in the registry determines the order used to search for valid licenses. This is most useful if you are mixing nodelocked and concurrent licenses on the same system. In this case, some applications will attempt to check out a license from the license server even if it is intended to be nodelocked. By placing the path of the nodelocked licenses ahead of the concurrent license path, you can ensure that the nodelocked license is read first. This is also how you configure license-file redundancy; simply place the names of the license servers in order in the variable. If one server is unavailable or out of licenses, the application will automatically switch to the next available server.

When used as an environment variable, INGR_LICENSE_FILE uses the same values as it does in the registry. If you want the license to be used by only one user, then set the variable in the USER environment settings for the specific user. If you use the SYSTEM environment settings, the license will be available to all users on the client system. If you plan to use the early-return feature in borrowing, INGR_LICENSE_FILE must be set as an environment variable.

If you have set INGR_LICENSE_FILE in both the environment and the registry, the environment variable is used first, and the registry value is appended to it.

To verify that the license file is installed correctly, initialize the application, and see if it runs correctly. If you get a dialog box indicating that you still need to get a license, then there could be either a problem with the license file or a problem contacting the license server.

3.2 License Borrowing

One drawback to concurrent licensing occurs when a user has to leave the network and still be able to use one of the licenses from the pool. This can be for any reason, but, typically, users may want to take the application with them when they travel or even when they go home for the weekend. This is where the ability to borrow concurrent licenses becomes important.

Borrowing allows a user to check out a license from the pool for a specified period of time. The maximum allowed time for borrowing Hexagon Safety & Infrastructure/Hexagon Geospatial application licenses is 2160 hours (90 days); however, the license administrator can further limit the amount of time available using the options file. Within these limitations, the user controls the amount of time they wish to use the borrowed license.

Borrowing is initiated on the client, no server side action is necessary. Once borrowing is initiated, the user specifies the date a borrowed license is to be returned and runs the application while connected to the network. The license is checked out from the server and is written to the client computer with the time remaining until expiration. Once the license has been checked out, the user is free to disconnect from the network. The client application will continue to use the borrowed license until the time expires (even if reconnected to the network). At the same time, the license server keeps the borrowed license checked out. When the borrowed time expires, the client application will require network access to get a new license, and the server will automatically make the borrowed license available again.

Note: Borrowing is user specific. Only the user who borrows the license can use the license. Other users on the same system will not have access to the borrowed license.

If a user is finished with the license before the borrowing period expires, that user has the option of returning the license early and making it available for other users. Only the user who borrowed the license can return it early. The license server has no control here and cannot force an early return. If the client containing the borrowed license is disabled or suffers a system failure, the license cannot be recovered. In this case, the license will remain in use until the borrow period expires.

No clock synchronization is required between the license server and the client. Borrowing is not date dependent, and altering a date on a system will not change the borrowing period.

Note: If you are using an expiring license rather than a permanent license, changing your system date by more than 24 hours will completely disable the licensing process. FlexNet Publisher considers this an attempt at tampering and will cause all licensing to fail. Permanent licenses will have the permanent key word included in the INCREMENT section.

Borrowing requires a specific type of concurrent license, and not all applications support the capability. Check your specific application for the borrowing capability, or contact your application support group. A concurrent license that allows for borrowing will contain the BORROW keyword. An example is shown below:

```
INCREMENT ISSD INGR 6.2 permanent 5 CDD11DFA7CEE BORROW=2160
ISSUER="Hexagon" ISSUED=10-Jan-2012 NOTICE="Hexagon"
```

There are three ways to initiate borrowing:

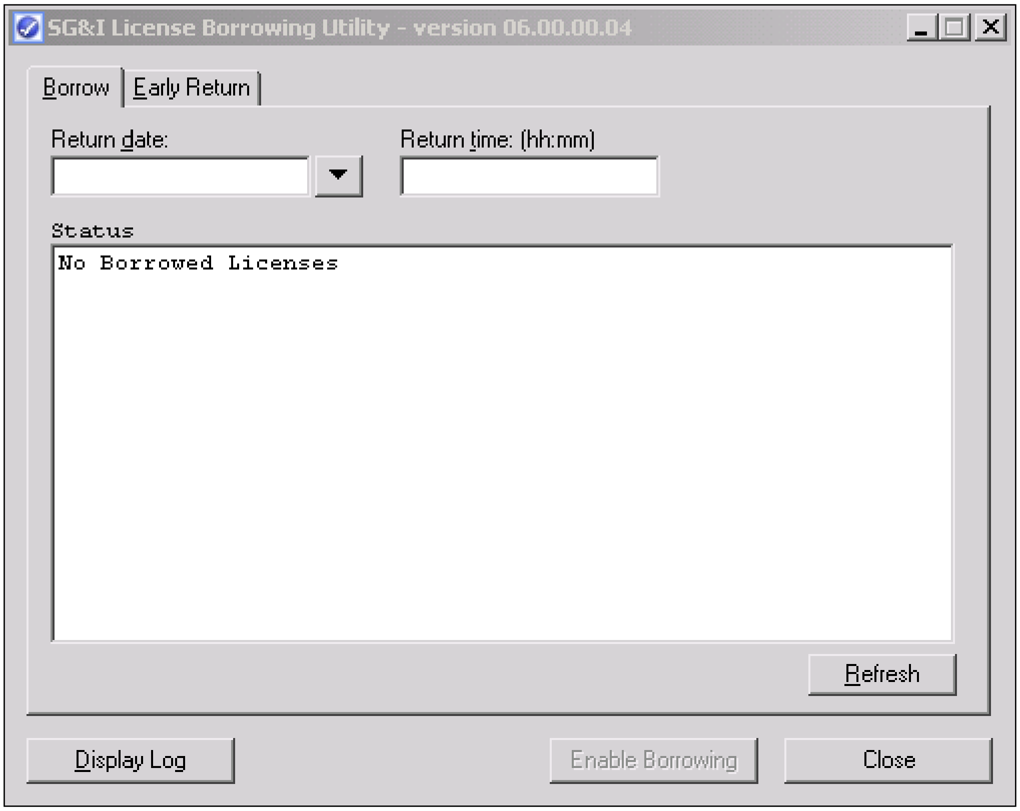
- Borrow Licenses utility – All Hexagon Safety & Infrastructure/Hexagon Geospatial applications
- LMTOOLS – FlexNet Publisher-based interface for managing licenses
- lmborrow – FlexNet Publisher command line application to initiate borrowing

3.2.1 SG&I License Borrowing Utility

To simplify the Borrowing and Early Return process, use the SG&I License Borrowing Utility. This is strictly a client side tool.

To install the **License Borrowing Utility** application, copy the contents of the *BorrowUtility.zip* file to a local directory. Once installed, you can use the following workflow to borrow a license:

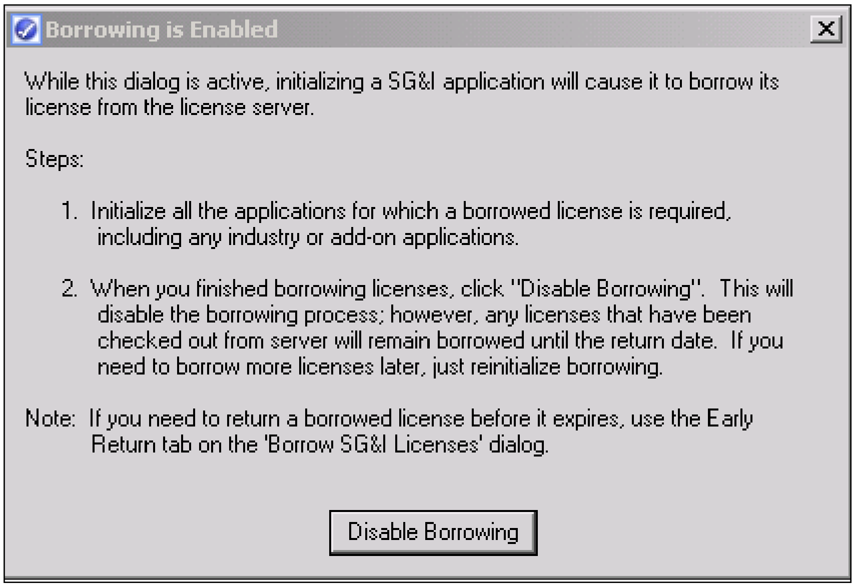
1. Initialize the **License Borrowing Utility** , and you should see the dialog box shown below:



No Borrowed Licenses

2. Set a Return date and, optionally, a Return time (use 24-hour military time, for example, 13:00 for 1 PM). If you do not set a return time, the license will be returned at 23:59 AM on the Return Date.

3. Select the **Enable Borrowing** button. This enables the borrowing process. The following dialog will be displayed:



Borrowing is Enabled

While this dialog is active, initializing a SG&I application will cause it to borrow its license from the license server.

Steps:

1. Initialize all the applications for which a borrowed license is required, including any industry or add-on applications.

2. When you finished borrowing licenses, click "Disable Borrowing". This will disable the borrowing process; however, any licenses that have been checked out from server will remain borrowed until the return date. If you need to borrow more licenses later, just reinitialize borrowing.

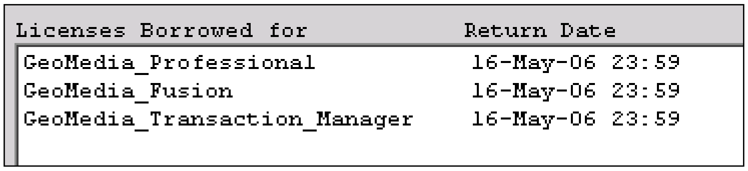
Note: If you need to return a borrowed license before it expires, use the Early Return tab on the 'Borrow SG&I Licenses' dialog.

   a. Follow the instructions on the **Borrowing is Enabled** dialog. While this dialog is active, initializing a Hexagon Safety & Infrastructure/Hexagon Geospatial application will cause it to borrow its license from the license server.

      Initialize all the applications for which a borrowed license is required, including any industry or add-on applications.

   b. When you finished borrowing licenses, click **Disable Borrowing**. This will disable the borrowing process; however, any licenses that have been checked out from the license server will remain borrowed until the return date.

   c. When you return to the main **License Borrowing Utility** dialog, the applications initialized while borrowing was enabled should appear in the list along with its return date and time.



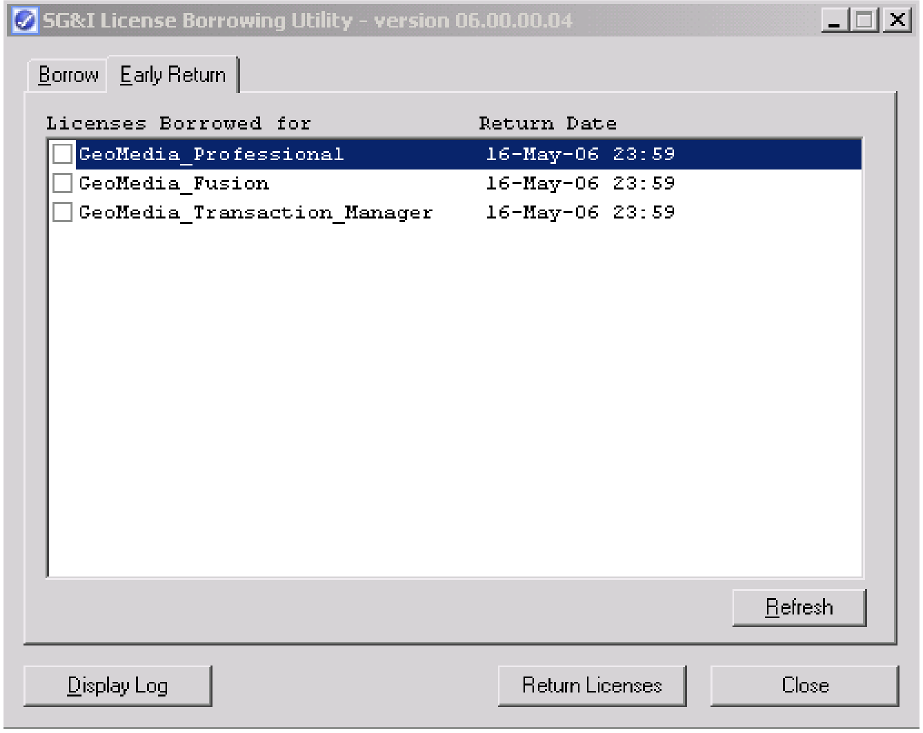
Licenses Borrowed for	Return Date
GeoMedia_Professional	16-May-06 23:59
GeoMedia_Fusion	16-May-06 23:59
GeoMedia_Transaction_Manager	16-May-06 23:59

At this point you can disconnect from the network and use your borrowed licenses. The listed applications will function normally until the return date.

If, for any reason, you need to return a license early, make sure you have reconnected to the network (and are recognized) and then perform the following steps:

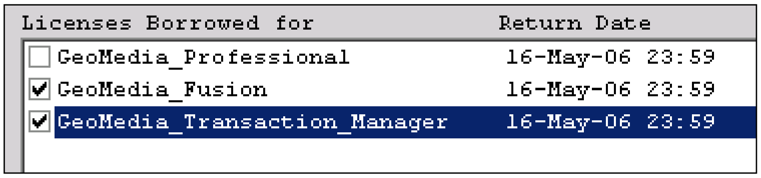
1. Initialize the **License Borrowing Utility.** The licenses you have already borrowed should appear in the list.

2. Switch to the **Early Return** tab. The dialog will look similar to the one below and you will see a list of the borrowed licenses eligible for early return:



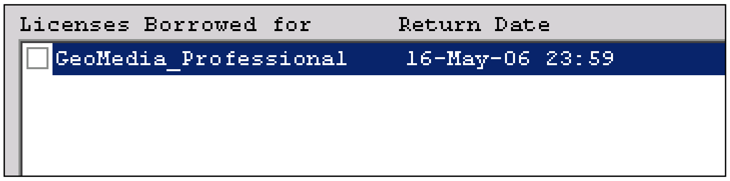
Licenses Borrowed for	Return Date
□ GeoMedia_Professional	16-May-06 23:59
□ GeoMedia_Fusion	16-May-06 23:59
□ GeoMedia_Transaction_Manager	16-May-06 23:59

3. Select the application license that you want to return (you can select more than one):



Licenses Borrowed for	Return Date
□ GeoMedia_Professional	16-May-06 23:59
☑ GeoMedia_Fusion	16-May-06 23:59
☑ GeoMedia_Transaction_Manager	16-May-06 23:59

4. Select the **Return Licenses** button. The dialog should refresh and show the currently borrowed licenses, the ones that were checked should be returned.



Licenses Borrowed for	Return Date
□ GeoMedia_Professional	16-May-06 23:59

5. If for any reason, one of the licenses was not returned, check the log file by selecting the **Display Log** button. Occasionally, there will be a network issue and

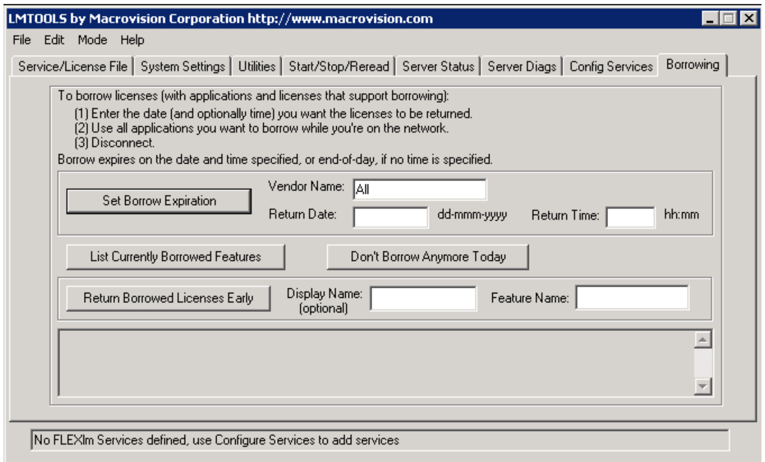
the early return will fail. The typical error is:

lmborrow: Cannot connect to license server system. (-15,570)

6. If you see this message, try the early return process again. If you need to return more licenses later, you can always run the early return process again.

3.2.2 Borrowing with LMTOOLS

The LMTOOLS utility has a borrowing option built in. You can use this instead of the License Borrowing utility if you want. To use the LMTOOLS borrowing option, copy the lmtools.exe utility to the client system where you want to initiate borrowing, and then run the process. Select the Borrowing tab, and follow the instructions given on the dialog box:



To borrow licenses (with applications and licenses that support borrowing):
(1) Enter the date (and optionally time) you want the licenses to be returned.
(2) Use all applications you want to borrow while you're on the network.
(3) Disconnect.
Borrow expires on the date and time specified, or end-of-day, if no time is specified.

Vendor Name: All

No FLEXlm Services defined, use Configure Services to add services

1. On the Borrowing tab, fill out the information regarding the vendor (INGR) and the license return date and time. When complete, click Set Borrow Expiration.

2. At this point, the borrowing capability is initiated. Any Hexagon Safety & Infrastructure/Hexagon Geospatial application that is initialized while this option is on will borrow a license. Initialize all the applications you want to borrow a license for. This is important, a license cannot be borrowed unless it is checked out by the application.

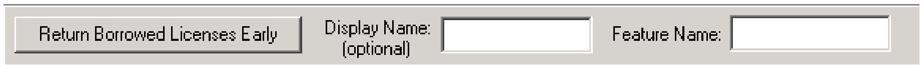
3. When you have initialized all the applications you want licenses for, click Don’t Borrow Anymore Today. This turns off the borrowing, but any licenses that have been borrowed will remain borrowed until the borrow period expires.

4. When you disconnect from the network, you should verify that the required licenses are actually borrowed by running the application.

5. You can also check the licenses that are currently borrowed by selecting the List Currently Borrowed Features button..

To return a license early, you must be connected to the network where you borrowed the license, and the note on page 24 above still applies. To return a license early using lmtools.exe, do the following:

1. Initialize LMTOOLS on the client, and select the Borrowing tab
2. Click List Currently Borrowed Features.
3. Pick the feature name you want to return from the list of available features, and enter it in the Feature Name field next to the Return Borrowed Licenses Early button.



4. Click Return Borrowed Licenses Early to return the license.
5. Repeat the process for every feature (application) you want to return.

Note: For early return to work using lmtools, you must set a local user variable INGR_LICENSE_FILE that points to the license server. If you do not do this, early return will fail.

3.2.3 Borrowing with lmborrow

FlexNet Publisher also provides a command line utility called lmborrow that facilitates borrowing and early return. These commands must be run in a command window. The following gis the syntax to enable the borrowing process:

*LMBORROW INGR dd-mmm-yyyy [hh:mm],*where dd-mmm-yyyy [hh:mm] is the return date and time. For example:

```
LMBORROW INGR 31-MAY-2006 08:00
```

While borrowing is enabled, initializing a Hexagon Safety & Infrastructure/Hexagon Geospatial application will cause it to borrow its license from the license server. If you want to see what you have currently borrowed, run the following:

```
LMBORROW -status
```

To stop the borrowing process, enter the following:

```
LMBORROW -clear
```

The syntax for early return is *LMBORROW –return feature.* To return a borrowed license early for GeoMedia Professional, you would enter the following:

```
SET INGR_LICENSE_FILE=@myserver
LMBORROW –return GeoMedia_Professional
```

Note: Setting INGR_LICENSE_FILE is required or early return will fail.

4. Troubleshooting

Even if you have a valid license configured on the server, you may still run into problems. This section is meant to help you troubleshoot some of the most common issues. If you need to contact customer support regarding a license problem, make sure you provide the specific error message received.

4.1 Standard Errors

The two most common errors that occur when initializing Hexagon Safety & Infrastructure/Hexagon Geospatial applications are as follows:

- License not found
- Invalid license file

With concurrent licenses in general, the two most common problems are:

- Licenses are checked out but are not checked back in.
- Early Return fails

4.1.1 LICENSE NOT FOUND

This error commonly occurs if the application cannot find the license file. This is generally caused by a server that is down or by an incorrect path to the license file. Other causes are as follows:

- If the license file is local, check the permissions on the license file folder. Users must be able to read the license file.
- Check the environment variables for the global license location variable LM_LICENSE_FILE or for the vendor-specific variable INGR_LICENSE_FILE. If either of these variables exists, they will override all other specified locations. If this is the case, you can solve this problem by appending the location of the license file to the existing path (use a semicolon as a separator). These keys are not typically used.
- If the environment variables do not exist, check the registry under the key:

  ```
  HKEY_LOCAL_MACHINE\SOFTWARE\FlexNet Publisher License Manager
  ```

  And look for either the LM_LICENSE_FILE or the INGR_LICENSE_FILE registry entry. Verify that the correct path to the license is included in the variable definition. Hexagon Safety & Infrastructure/Hexagon Geospatial applications will automatically set the INGR_LICENSE_FILE to the default location:

```
C:\Program Files\Common Files\Intergraph\INGR_Licenses
```

if this path is incorrect or if you want to use another path, make the change here.

- For expiring licenses, changing the system date by more than 24 hours can also cause this error. FlexNet Publisher has built-in checks for tampering and can determine if the system date has been altered. If this is suspected to be the cause of the problem, you will need to find and correct the affected file. The only way to do this is to search the system for any files with a creation or modification date greater than today's date. If any files are found, they will need to be set back to today's date.

NOTE: For additional information on the License File Not Found error message, please review the document "Analyzing Errors" located on the license generation web site, https://licensing.hexagonsafetyinfrastructure.com/flexnet/operationsportal/.

4.1.2 INVALID LICENSE FILE

If the path information is correct and the application can read the license file, the next most common error is an invalid license file. This error indicates that there is something wrong with the license file itself. Typical problems here include the following:

- The entry on the SERVER line *this_host* has not been changed to reflect the TCP/IP address or the actual name of the license server for licenses using the static TCP/IP address. The easiest way to solve this is to ensure that *this_host* is set to the same TCP/IP address used in the INTERNET=.

- For COMPOSITE Host IDs, modify the entry *this_host* to the actual name of the license server.

- The INTERNET=###.###.###.### value on the SERVER line does not match the primary TCP/IP address of the license server. If this is the case, move all the concurrent license files to the correct server. If the address is in error, you will need to rehost all the affected licenses with the correct TCP/IP address. You may rehost the license files online (see detailed instructions in the Help topics on the License Generation website) or complete a Rehost Request form and email the completed form to license@hexagonsi.com.

- For COMPOSITE HOST IDs, you may not have the correct DNS name in the *this_host* entry.

- The INCREMENT sections of the license pool file have been modified or are not correct for the application returning the error. Verify that the license information is correct. If you are unsure, you will need to contact license@hexagonsi.com.

- The license file was not created correctly from the license web site. Again, you will need to rehost all the affected licenses with the correct address. You may rehost the license files online (see detailed instructions in the Help topics on the License Generation website) or complete a Rehost Request form and email the completed form to license@hexagonsi.com.

- To help to further diagnose the problem, try these steps using LMTOOLS:

  1. On the license server, run LMTOOLS.

  2. On the main dialog, select the **Server Diags** tab.

  3. Click **Perform Diagnostics**. This will check the status of every license in the pool file. Valid licenses will look like the following:

     ```
     License file: 27000@myserver
     -----------------------------------------------------
     "GeoMedia_Professional" v6.1, vendor: Integraph
     License server: 129.139.149.159
     floating license

     This license can be checked out
     -----------------------------------------------------
     ```

     Invalid entries will either be skipped or will return an error.

  4. Another log file that can generate useful information can be found under the **Config Services** tab.

  5. Click **View Log** to view the general usage log. Error messages and error codes will also show up in this file and may provide some insight into what the problem is.

If you do not see any obvious problems in the log files, you can always open a worksheet with Hexagon Safety & Infrastructure/Hexagon Geospatial Customer Support and pass this information to them.

4.1.3 License is Not Checked Back into the License Pool

This problem occurs when a client is no longer using a license that has been checked out from the license pool but the licenses is never checked back in. This can happen if the process that checks the license in fails. An application crash or process hanging may contribute to this. To solve this problem, make use of the *INGR.opt* file and set a TIMEOUT or TIMEOUTALL parameter:

```
TIMEOUTALL 300
```

This will ensure that the license is checked back in automatically if a heartbeat is not received from the client application. See section 2.6 for more information.

4.1.4 Early Return is failing

The most common reason why early return fails is due to the local environment variable not being set. For early return to work, INGR_LICNESE_FILE must be set as a local user variable and must point to the license server. The entry in the registry will not suffice here.

Another common cause for early return failure is a network problem. The typical error is:

lmborrow: Cannot connect to license server system. (-15,570)

Sometimes network traffic causes this and all you need to do is try the return process again. If you continually get this problem, you need to verify your network connectivity and whether you can still access the license server.

4.2 Diagnostics

You have the ability to collect diagnostic information on the issue before contacting Customer Support. Create an environment variable called FLEXNET PUBLISHER_DIAGNOSTICS, and set the value of the variable to 1, 2, or 3, depending on the level of diagnostics that you desire. This variable will generate diagnostic information whenever a license checkout is denied. The output is a file in the license directory called *flexpid.log.*

4.2.1 Level 1 Diagnostics

If FLEXNET PUBLISHER_DIAGNOSTICS is set to 1, then the standard FlexNet Publisher error message is displayed along with a complete list of license files that the application tried to use. For example:

```
FlexNet Publisher checkout error: Cannot find license file (-1,73:2)
No such file or directory license file(s): C:\IngrLicense\Ingr.lic
```

4.2.2 Level 2 Diagnostics

If FLEXNET PUBLISHER_DIAGNOSTICS is set to 2, the arguments used to check out licenses are presented along with the level 1 information. For example:

```
FlexNet Publisher checkout error: No such feature exists (-5,116:2) No
such file or directory license file(s): C:\IngrLicense\Ingr.lic
lm_checkout("GeoMedia", 6.0, 1, 0x0, ..., 0x4000)
```

In the example above, the error message contains the following two separate problems, both of which occurred during the license checkout:

- The feature does not exist in the license file that was found.
- The other license file was not found.

The message *lm_checkout* contains the following information:

```
lm_checkout(feature, version, num_lic, queue_flag, ..., dupgroup_mask)
```

where:

feature – The requested feature.
version – The requested version.

num_lic – The number of licenses requested, usually 1.
queue_flag – 0 - no queuing; 1 - queue for license; 2 - queue for licenses
dupgroup_mask – Indicates duplicate grouping, also called license sharing.

4.2.3 Level 3 Diagnostics

If FLEXNET PUBLISHER_DIAGNOSTICS is set to 3, then, in addition to the information provided by level 1 and 2, an explanation of how the licenses were granted is also included:

```
Checkout succeeded: f0/BC64A7B120AE
License file: INGR.lic License Server: @hitech
```

For most cases, level 2 diagnostics is sufficient.

5. Other Information

For more information on concurrent licensing or licensing issues in general, there are several resources at your disposal:

- For Hexagon Safety & Infrastructure/Hexagon Geospatial licensing resources, check out the following: https://support.hexagonsafetyinfrastructure.com/infocenter/index?page=licensing_resources

Hexagon Safety & Infrastructure/Hexagon Geospatial offers a wide variety of support resources:

- General support information can be found at the following: http://support.hexagonsafetyinfrastructure.com/
- The Hexagon Safety & Infrastructure/Hexagon Geospatial Knowledge Base offers information on a wide variety of topics. You can access the Knowledge Base at the following: https://support.hexagonsafetyinfrastructure.com/infocenter/index?page=licensing_resources