



TASKING License Management

Copyright © 2025 TASKING B.V.

All rights reserved. You are permitted to print this document provided that (1) the use of such is for personal use only and will not be copied or posted on any network computer or broadcast in any media, and (2) no modifications of the document is made. Unauthorized duplication, in whole or part, of this document by any means, mechanical or electronic, including translation into another language, except for brief excerpts in published reviews, is prohibited without the express written permission of TASKING B.V. Unauthorized duplication of this work may also be prohibited by local statute. Violators may be subject to both criminal and civil penalties, including fines and/or imprisonment. TASKING® and its logo are registered trademarks of TASKING Germany GmbH. All other registered or unregistered trademarks referenced herein are the property of their respective owners and no trademark rights to the same are claimed.

Table of Contents

Manual Purpose and Structure	v
1. Licensing Overview	1
1.1. Licensing Terminology	1
1.2. License Types	3
2. Licensing TASKING Products	7
2.1. Licensing winIDEA Products	7
2.2. Obtaining a License	7
2.3. Frequently Asked Questions (FAQ)	8
2.4. Installing a License	9
2.4.1. Configure the Firewall in your Network	9
2.4.2. Installing Server Based Licenses (Floating or Node-Locked)	9
2.4.3. Installing Client Based Licenses (Node-Locked)	11
2.5. License Files Created During Installation	13
2.5.1. Licensing Options File	13
2.5.2. License File	18
3. Local License Server	19
3.1. Overview	19
3.2. Replacing the Local License Server with a Newer Version	21
3.3. Installing a Local License Server	21
3.3.1. Start the Local License Server as a Service on Windows	23
3.3.2. Start the Local License Server as a Daemon on Linux	23
3.3.3. Start the Local License Server from the Command Line	24
3.3.4. Stop and Delete the TASKING License Service on Windows	24
3.3.5. Stop and Remove the taskinglm Service on Linux	25
3.3.6. Obtaining a License for a Local License Server	25
3.4. Local License Server Options and Defines	26
3.5. Redundant Master - Slave Configuration	28
3.6. Creating a Local License Server License File	30
3.7. License Management Event Log	31
4. Administering Licenses	35
4.1. Monitoring the Status of the Local License Server	35
4.2. Transferring a License (Borrow Feature)	38
4.3. Joining or Grouping Licenses by Using a License Pool	41
4.4. Wait for a Free License	44
4.5. Using a License Queue	45
4.6. Bulk Installation	46
4.7. Restricting Access to Licenses	46

Manual Purpose and Structure

Manual Purpose

You should read this manual if you:

- install a licensed TASKING product and need more information about licensing and installation than what is described in the Getting Started manual of the TASKING product
- are upgrading or otherwise modifying existing licenses
- install a TASKING local license server on a network

Manual Structure

Chapter 1, Licensing Overview

Contains an overview of the terminology used in this manual and describes the different license types.

Chapter 2, Licensing TASKING Products

Explains step-by-step how to obtain and install a license for a TASKING product. Also contains a list of options and environment variables that a TASKING product uses to check for a license. Furthermore it explains the use of a license file.

Chapter 3, Local License Server

Explains how to install and configure a local license server.

Chapter 4, Administering Licenses

Explains how to monitor the status of the local license server. Also explains how to borrow, join or bulk install licenses, use license pools and the license queue and control access to licenses.

Related Publications

For details about installing a TASKING product, see the Getting Started manual delivered with the product.

Chapter 1. Licensing Overview

This chapter contains an overview of the licensing terminology used and the available license types.

1.1. Licensing Terminology

TASKING products are protected with TASKING license management (TLM) software. To use a TASKING product, you must install that product and install a license.

The following terminology is used to describe the licensing concepts and software components (in alphabetical order).

Access file

An access file is a file used by a local license server with rules to bind, deny and permit users and systems to licenses.

Client

A client is a TASKING product for which a license is needed. For example, a C compiler.

Computer ID

The computer ID is an alphanumeric value that uniquely identifies your computer. The Computer ID is required to identify the computer for a node-locked license.

Heartbeat

The heartbeat is the duration in seconds after which a client tries to refresh a license from the server.

License

A license is the right to use a product. During the execution of a TASKING product (client), the client verifies whether you have the right to use it. Therefore, the client sends the license key to the license server. If you are allowed to use the client the license server sends a granted license back to the client. If you use a node-locked client based license the license is not retrieved from a license server but is read from a license file stored on your computer.

License key

The license key is a unique key (in the format xxxx-xxxx-xxxx-xxxx) provided to you by TASKING. The license key is associated with the product you ordered, it identifies that you have the right to use the product. During the installation of your product you need to enter this key.

TASKING License Management

License pool

A license pool is a pool of licenses from which the local license server can pick a license. By using a license pool you can join or group licenses for a TASKING product.

License server

A license server is the process that handles the license requests from client processes. A license server can be a remote license server on the internet or a local license server on your company network.

In this manual "TASKING Local License Server" and "Local License Server" are used as synonyms. "Remote License Server" is used as a synonym for "TASKING Remote License Server".

License type

Any of the license types as explained in [Section 1.2, License Types](#).

Linger time

The linger time is a period in which the license server recognizes the license as being in-use after the user exits the software and the license is no longer running. During that period another user cannot use the license. After the linger time has finished, the license seat will be freed by the license server.

In build systems or larger builds that employ multiple files to be compiled and linked, the build time is such that it may happen that another client attempts to use a license. As the tools (compiler, assembler) are individually called per file to compile and assemble, a different user may 'grab' a license in between calls and the build would terminate abnormally. To avoid this, especially if the number of available licenses is exhausted, the linger time was introduced.

Pool name

A pool name is the name of a license pool in the format POOL-xxxxx-xxxx-xxxx for TASKING products prior to March 2023, or POOL-*name* for TASKING products built with TLM v1.5r1 or newer.

Process limit

The maximum number of parallel processes allowed by a license.

Queue

Starting with v2.2r1, the local license server supports a license queue mechanism. When all seats are in use, a license request can be put in a queue. This works on a First In First Out (FIFO) basis. When a free seat becomes available, the first one waiting in the queue will get the license. You can turn license queuing on globally or per license. See [Section 4.5, Using a License Queue](#) for more information.

Seats

The number of users that are allowed to use the client according to the license.

Security ID

The security ID is an alphanumeric value that uniquely identifies the computer that runs a local license server.

SLA

The [Software License Agreement](#).

1.2. License Types

The following license types can be ordered from TASKING.

Node-locked license

A node-locked license locks the software to one specific computer so you can use the product on that particular computer only.

For information about installing a node-locked license see [Section 2.4.2, *Installing Server Based Licenses \(Floating or Node-Locked\)*](#) and [Section 2.4.3, *Installing Client Based Licenses \(Node-Locked\)*](#).

Floating license

A floating license is a license located on a license server and can be used by multiple users on the network. Floating licenses allow you to share licenses among a group of users up to the number of users (seats) specified in the license.

For example, suppose 50 developers may use a client but only ten clients are running at any given time. In this scenario, you only require a ten seats floating license. When all ten licenses are in use, no other client instance can be used. Also a linger time is in place. This means that a license seat is locked for a period of time after a user has stopped using a client. The license seat is available again for other users when the linger time has finished.

For information about installing a floating license see [Section 2.4.2, *Installing Server Based Licenses \(Floating or Node-Locked\)*](#).

Build server license

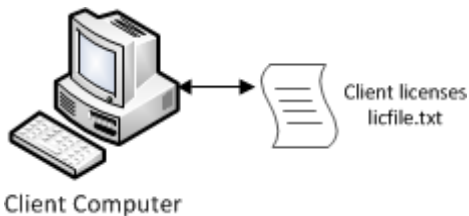
A build server license is a license on a central computer system, shared (used) by other systems, automated processes or users. A build server license is limited to a maximum number of parallel processes per user (seat). A build server license is usually installed as a floating license with a process limit.

When the maximum number of parallel processes is reached a new client process is set to waiting until the number of running client processes under the same license seat is below the process limit.

License service types

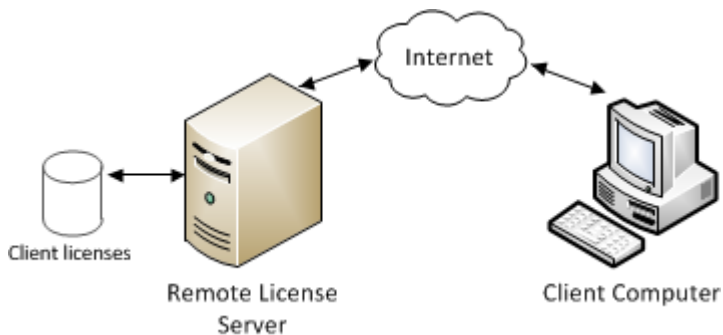
The license service type specifies the process used to validate the license. The following types are possible:

- **Client based** (also known as 'standalone'). The license is serviced by the client. All information necessary to service the license is available on the computer that executes the TASKING product. This license service type is available for node-locked licenses only.

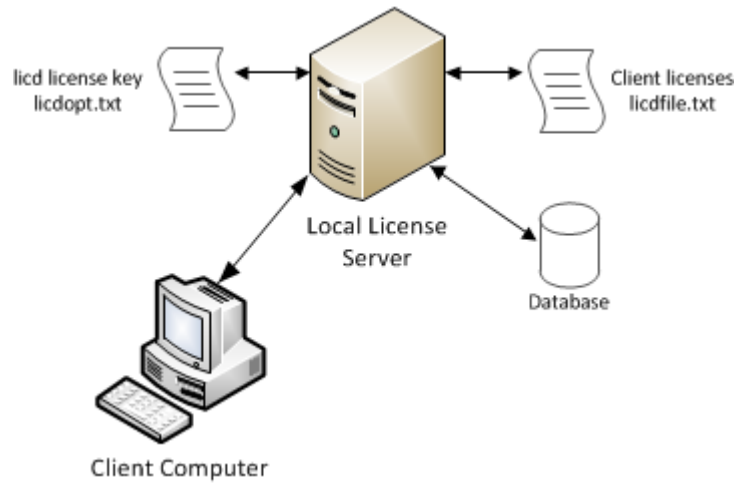


- **Server based** (also known as 'network based'). The license is serviced by a separate license server program that runs either on your companies' network or runs in the cloud. This license service type is available for both node-locked licenses and floating licenses.

Licenses can be serviced by a cloud based license server called "**Remote License Server**". This is a license server that is operated by TASKING.



Alternatively, you can install a license server program on your local network. Such a server is called a "**Local License Server**". You have to configure such a license server yourself, see [Section 3.3, Installing a Local License Server](#).



The benefit of using the Remote License Server is that product installation and configuration is simplified.

Unless you have an IT department that is proficient with the setup and configuration of licensing systems we recommend to use the facilities offered by the Remote License Server.

TASKING License Management

Chapter 2. Licensing TASKING Products

This chapter guides you through the process of licensing a TASKING product.

Installation advice when using a local license server

During the installation of a TASKING product you can specify to use a local license server. If you do so, the installation program asks for the server name and port number of the local license server. If you do not know this information, you can simplify the installation if you first install and run the local license server before you install the TASKING product. See [Section 3.3, *Installing a Local License Server*](#).

2.1. Licensing winIDEA Products

The installation and licensing of winIDEA products differs from other TASKING products that are licensed with TASKING License Management. For details follow the instructions in the license email. For more information see <https://www.isystem.com/downloads/winIDEA/help/license-request.html>

2.2. Obtaining a License

You need a license key when you install a TASKING product on a computer. If you have not received such a license key follow the steps below to obtain one. Otherwise, you cannot install the software.

Obtaining a server based license (floating or node-locked)

- Order a TASKING product from TASKING or one of its distributors.

A license key will be sent to you by email or on paper.

If your node-locked server based license is not yet bound to a specific computer ID, the license server binds the license to the computer that first uses the license.

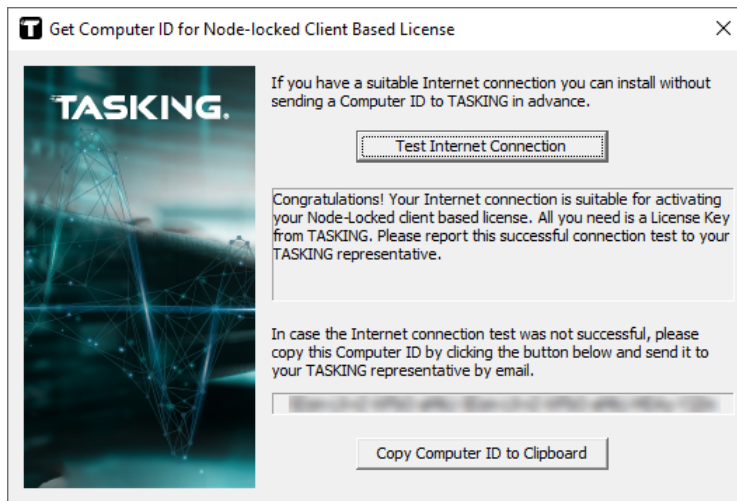
Obtaining a client based license (node-locked)

To use a TASKING product on one particular computer with a license file, TASKING needs to know the computer ID that uniquely identifies your computer. You can do this with the **getcid** program that is available on the TASKING website. The detailed steps are explained below.

1. Download the **getcid** program from <https://www.tasking.com/support/tlm/downloads>.
2. Execute the **getcid** program on the computer on which you want to use a TASKING product. The tool has no options. For example,

Windows (GUI):

```
getcid_version
```



The computer ID is displayed in the lower part of the dialog.

Windows (command prompt):

```
getcidcmd_version
```

The computer ID is displayed on your screen.

Linux:

```
./getcid_version
```

```
ComputerID: 5Dzm-L9+Z-WFbO-aMkU-5Dzm-L9+Z-WFbO-aMkU-MDAy-Y2Zm
```

The computer ID is displayed on your screen.

3. Order a TASKING product from TASKING or one of its distributors and supply the computer ID.

A license key and a license file will be sent to you by email or on paper.

When you have received your TASKING product, you are now ready to install it.

2.3. Frequently Asked Questions (FAQ)

If you have questions or encounter problems you can check the TASKING Knowledge Base.

<https://kb.tasking.com/KB/83>

This page contains answers to questions for the TASKING license management system TLM.

If your question is not there, please contact your nearest TASKING Sales & Support Center or Value Added Reseller.

2.4. Installing a License

On Windows the license setup procedure is done by the installation program. On Linux you have to install a license manually.

If the installation program on Windows can access the internet then you only need the license key. Given the license key the installation program retrieves all required information from the remote license server. The installation program sends the license key and the computer ID of the computer on which the installation program is running to the remote license server, no other data is transmitted.

If the installation program cannot access the internet the installation program asks you to enter the required information by hand. If you install a node-locked client based license you should have the license file at hand (see [Section 2.2, Obtaining a License](#)).

Floating licenses are always server based and node-locked licenses can be server based. All server based licenses are installed using the same procedure.

2.4.1. Configure the Firewall in your Network

For using the TASKING license servers the TASKING license manager tries to connect to the remote license servers `lic1.tasking.com`, `lic2.tasking.com`, `lic3.tasking.com`, `lic4.tasking.com` at the TCP ports 8080, 8936 or 80. Make sure that the firewall in your network is transparently enabled for one of these ports.

2.4.2. Installing Server Based Licenses (Floating or Node-Locked)

If you do not have received your license key, read [Section 2.2, Obtaining a License](#) before you continue.

Windows

1. If you want to use a local license server, first install and run the local license server before you continue with step 2. See [Section 3.3, Installing a Local License Server](#).
2. Install the TASKING product and follow the instructions that appear on your screen.

The installation program asks you to enter the license information.

TASKING License Management

The screenshot shows a dialog box titled "TASKING <product> - InstallShield Wizard". The main heading is "License Key Information" with the instruction "Specify your license key". The TASKING logo is in the top right. Below the heading, there is a paragraph: "Please enter the license key that you have received from TASKING. The key has the format like aaa-bbbb-cccc-dddd. If you do not have a key, please contact TASKING through licensing@tasking.com, or contact your TASKING representative." A text input field labeled "License Key:" is provided. At the bottom, there are four buttons: "Licensing Support", "< Back", "Next >", and "Cancel".

3. In the **License Key** field enter the license key you have received from TASKING and click **Next** to continue.

*The installation program tries to retrieve the license information from a remote license server. Wait until the license information is retrieved. If the license information is retrieved successfully subsequent dialogs are already filled-in and you only have to confirm the contents of the dialogs by clicking the **Next** button. If the license information is not retrieved successfully you have to enter the information by hand.*

4. Select your **License Type** and click **Next** to continue. If the license type is already filled in and grayed out, you can just click **Next** to continue.

You can find the license type in the email or paper that contains the license key.

5. (For floating licenses only) Select **Remote license server** to use one of the remote license servers, or select **Local license server** for a local license server. The latter requires optional software.

(For local license server only) specify the **Server name** and **Server port** of the local license server.

6. Click **Next** and follow the rest of the instructions to complete the installation.

Linux

1. If you want to use a local license server, first install and run the local license server before you continue with step 2. See [Section 3.3, Installing a Local License Server](#).
2. After installation of the TASKING product, the license options file `licopt.txt` is present in the directory `<install-dir>/<product-name>/etc`, for example `/opt/TriCore/etc` for the TriCore product, assuming you have installed the TASKING product in `/opt`. Edit this file with a text editor and change the line:

```
TSK_LICENSE_KEY_SW160800 = 0000-0000-0000-0000
```


Replace 0000-0000-0000-0000 with the license key you obtained from TASKING.

3. Set the environment variable `TSK_OPTIONS_FILE_<Product-Code><version>` to the absolute location of `licopt.txt`. For example:

```
TSK_OPTIONS_FILE_SW160800v6_3r1=/opt/TriCore/etc/licopt.txt
export TSK_OPTIONS_FILE_SW160800v6_3r1
```

2.4.3. Installing Client Based Licenses (Node-Locked)

If you do not have received your license key and license file, read [Section 2.2, Obtaining a License](#) before continuing.

Windows

1. Install the TASKING product and follow the instructions that appear on your screen.

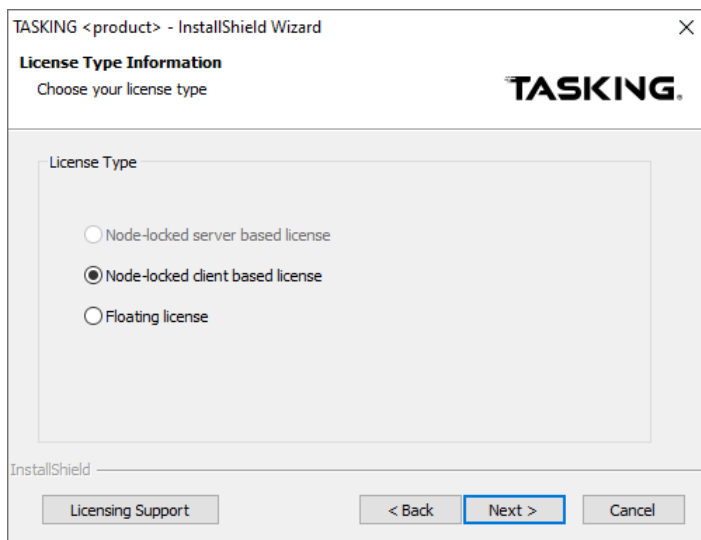
The installation program asks you to enter the license information.

The screenshot shows a dialog box titled "TASKING <product> - InstallShield Wizard". The main heading is "License Key Information" with the instruction "Specify your license key". The TASKING logo is displayed in the top right. Below the heading, there is a text block: "Please enter the license key that you have received from TASKING. The key has the format like aaa-bbbb-cccc-dddd. If you do not have a key, please contact TASKING through licensing@tasking.com, or contact your TASKING representative." A text input field labeled "License Key:" is provided for the user to enter the key. At the bottom, there are four buttons: "Licensing Support", "< Back", "Next >", and "Cancel". The "InstallShield" logo is visible in the bottom left corner.

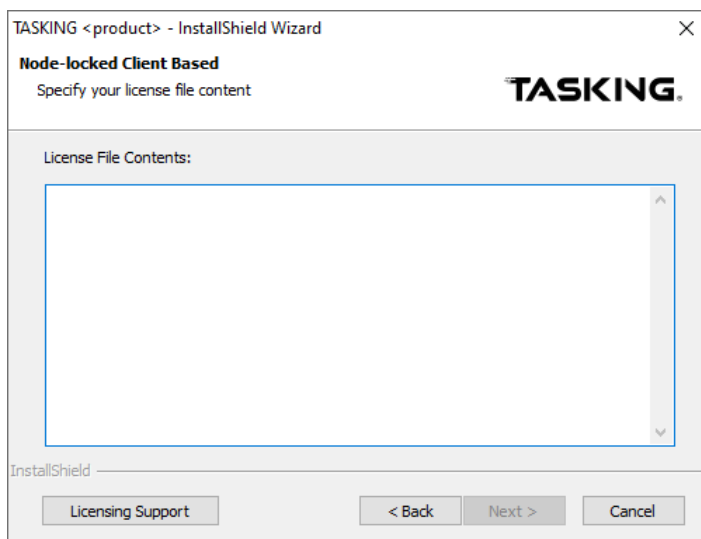
2. In the **License Key** field enter the license key you have received from TASKING and click **Next** to continue.

*The installation program tries to retrieve the license information from a remote license server. Wait until the license information is retrieved. If the license information is retrieved successfully subsequent dialogs are already filled-in and you only have to confirm the contents of the dialogs by clicking the **Next** button. If the license information is not retrieved successfully you have to enter the information by hand.*

TASKING License Management



3. Select **Node-locked client based license** and click **Next** to continue.



4. In the **License File Contents** field enter the contents of the license file you have received from TASKING.

The license data is stored in the file licfile.txt in the etc directory of the product (<install_dir>\etc).

5. Click **Next** and follow the rest of the instructions to complete the installation.

Linux

1. Install the TASKING product and put the license file `licfile.txt` you have received from TASKING in the directory `<install-dir>/<product-name>/etc`. For example, if you installed a TriCore product in `/opt`, the path is `/opt/TriCore/etc`
2. After installation of the TASKING product, the license options file `licopt.txt` is present in the directory `<install-dir>/<product-name>/etc`. Edit this file with a text editor:

- a. Change the line:

```
TSK_LICENSE_KEY_SW160800 = 0000-0000-0000-0000
```

Replace `0000-0000-0000-0000` with the license key you obtained from TASKING.

- b. Change the line:

```
#TSK_LICENSE_FILE_SW160800 = licfile.txt
```

to:

```
TSK_LICENSE_FILE_SW160800=/opt/TriCore/etc/licfile.txt
```

Specify the absolute path to the file `licfile.txt`.

3. Set the environment variable `TSK_OPTIONS_FILE_<Product-Code><version>` to the absolute location of `licopt.txt`. For example:

```
TSK_OPTIONS_FILE_SW160800v6_3r1=/opt/TriCore/etc/licopt.txt
export TSK_OPTIONS_FILE_SW160800v6_3r1
```

2.5. License Files Created During Installation

The installation program on Windows stores the license key in the file `licopt.txt` in the `etc` directory of the product. See [Section 2.5.1, Licensing Options File](#) for details about this file. Additionally, for a node-locked client base license only, the installation program stores the license data in the file `licfile.txt` in the `etc` directory of the product. See [Section 2.5.2, License File](#) for more information.

2.5.1. Licensing Options File

All information needed by the client program to perform the license check is read from the options file `licopt.txt` or from environment variables. The options in the options file have precedence over the environment variables.

You can specify the name and location of the options file in an environment variable. When you specify an absolute path then that file is taken. When you specify a relative path or only the filename then the specified file is searched for according to a search path.

The `#` character marks the start of a comment.

TASKING License Management

The license options file is searched for in this order:

1. Search for the license options file in an environment variable in this order:

```
TSK_OPTIONS_FILE_SWxxxxxxvX_YrZ=file
TSK_OPTIONS_FILE_SWxxxxxxhhXXY=file
TSK_OPTIONS_FILE_SWxxxxxx=file
TSK_OPTIONS_FILE=file
```

where,

SWxxxxxx is the product code,

hh is the host code (02 is Linux, 05 is Windows),

vX_YrZ is the full version of the product, and

XXY is the major (*XX*) and minor (*Y*) version of the product.

Examples:

```
TSK_OPTIONS_FILE_SW160800v6_3r1=C:\TriCore_v6.3r1\etc\licopt.txt (*)
TSK_OPTIONS_FILE_SW160800v6_3r1=/opt/TriCore_v6.3r1/etc/licopt.txt
TSK_OPTIONS_FILE_SW16080002062=/opt/TriCore_v6.2r2/etc/licopt.txt
TSK_OPTIONS_FILE_SW160800=sw160800-opt.txt (**)
TSK_OPTIONS_FILE=myoptions.txt (**)
```

(*) This variant is automatically set by the installation program of a TASKING product for Windows. For Linux you have to set the environment variable yourself.

(**) The files `sw160800-opt.txt` and `myoptions.txt` are searched for according to the rules specified in the search path below.

2. When no environment variable exists for the license options file, the default filename searched for is `licopt.txt` according to the following search path. This path is also searched for when you have specified a relative path in one of the environment variables.

Windows:

```
.;\etc;..\etc;tooldir\..\etc;tooldir\..\..\etc;tooldir\..\bin;tooldir
```

Linux:

```
./etc;../etc;tooldir/./etc;tooldir/../../etc;tooldir/./bin;tooldir
```

License options

The following options can be present in the license options file or as an environment variable:

Option / Environment Variable	Description	Mandatory / Optional
<p>TSK_LICENSE_KEY= <i>License-key</i> <i>pool-name</i> or TSK_LICENSE_KEY_SWxxxxxx= <i>License-key</i> <i>pool-name</i> or TSK_LICENSE_KEY_SWxxxxxxhhXXY= <i>License-key</i> <i>pool-name</i> or TSK_LICENSE_KEY_SWxxxxxxvX_YrZ= <i>License-key</i> <i>pool-name</i></p>	<p>Assign the license key <i>License-key</i> to a TASKING product. If you have several license keys for different products you can use the software product code in the name. The different variants are the same as specified above this table with TSK_OPTIONS_FILE. If a license pool name is entered instead of a license key, then the client can retrieve an arbitrary license associated with the TASKING product from a license pool from the local license server. This functionality is, for example, used for joining licenses (see Section 4.3, Joining or Grouping Licenses by Using a License Pool). You can find the software product code in the email or paper that contains the license key.</p> <p>When you specify an environment variable, matching is done in reverse order as listed in the table. When you have specified more than one option in the options file the last one matching in the file is taken.</p>	<p>Mandatory</p>
<p>TSK_LICENSE_SERVER= <i>host</i> [: <i>port</i>] [...]</p>	<p>Specifies the list of license servers and associated port numbers that can be contacted by a client. A client tries to connect to a server in the order specified in the list. Only if a server is offline then a connection to another (backup) server is made. The server can be a remote license server or a local license server. The default is:</p> <pre> lic1.tasking.com:8080; lic1.tasking.com:8936; lic1.tasking.com:80; lic3.tasking.com:8080; lic3.tasking.com:8936; lic3.tasking.com:80; lic2.tasking.com:8080; lic2.tasking.com:8936; lic2.tasking.com:80; lic4.tasking.com:8080; lic4.tasking.com:8936; lic4.tasking.com:80 </pre>	<p>Optional</p>

TASKING License Management

Option / Environment Variable	Description	Mandatory / Optional										
TSK_LICENSE_PORT= <i>port</i>	Connect to port <i>port</i> on the license server. For the remote license server you can use port 8080, 8936 or port 80 (the remote license server listens on all three ports). If you use a local license server, specify 9090 or the port that you used during the installation of the local license server.	Optional										
TSK_LICENSE_PROXY_SERVER= <i>proxyserver[:port]</i>	If a proxy server is required for outgoing license requests, use this option to specify <i>proxyserver</i> as the HTTP proxy server. If you omit the <i>port</i> , the default port is 3128.	Optional										
TSK_LICENSE_PROXY_PORT= <i>port</i>	Specify the TCP port number to use for a proxy server (default: 3128).	Optional										
TSK_LICENSE_FILE= <i>file</i> or TSK_LICENSE_FILE_SWxxxxxx= <i>file</i> or TSK_LICENSE_FILE_SWxxxxxxhhXY= <i>file</i> or TSK_LICENSE_FILE_SWxxxxxxX_YrZ= <i>file</i>	If this option or environment variable is set, the client retrieves a license from the license file <i>file</i> instead of a remote server. Use this option when you use a node-locked client-based license (default: <i>licfile.txt</i>). The search path is the same as for the options file. Alternatively, you can specify an absolute path. If you have several license files for different products you can use the software product code in the name. The different variants are the same as specified above this table with TSK_OPTIONS_FILE. When you specify an environment variable, matching is done in reverse order as listed in the table. When you have specified more than one option in the options file the last one matching in the file is taken.	Optional										
TSK_LOG_VERBOSE[= <i>level</i>]	Log verbose, repeat in the option file for more verbose, or specify the verbose level. With this option log information about a license request by a client is sent to <i>stderr</i> . This option can be useful for debugging licensing problems. You can specify the following levels: <table border="1" data-bbox="543 1298 1061 1584"> <thead> <tr> <th data-bbox="543 1298 617 1338">Level</th> <th data-bbox="617 1298 1061 1338">Description</th> </tr> </thead> <tbody> <tr> <td data-bbox="543 1338 617 1378">0</td> <td data-bbox="617 1338 1061 1378">No verbose logging.</td> </tr> <tr> <td data-bbox="543 1378 617 1418">1</td> <td data-bbox="617 1378 1061 1418">License errors.</td> </tr> <tr> <td data-bbox="543 1418 617 1458">2</td> <td data-bbox="617 1418 1061 1458">General license information.</td> </tr> <tr> <td data-bbox="543 1458 617 1584">3 or higher</td> <td data-bbox="617 1458 1061 1584">Detailed license configuration information, useful for debugging. For example, the environment variables or options used, the license key, the search path.</td> </tr> </tbody> </table>	Level	Description	0	No verbose logging.	1	License errors.	2	General license information.	3 or higher	Detailed license configuration information, useful for debugging. For example, the environment variables or options used, the license key, the search path.	Optional
Level	Description											
0	No verbose logging.											
1	License errors.											
2	General license information.											
3 or higher	Detailed license configuration information, useful for debugging. For example, the environment variables or options used, the license key, the search path.											

Option / Environment Variable	Description	Mandatory / Optional
TSK_LICENSE_WAIT[= <i>sec</i>]	Wait for a free license. With this option you can set the license retry time-out (default: none). When a request for a floating license is denied because no free seats are available or the license server is unresponsive then the license request(s) will be retried for a maximum of <i>sec</i> seconds or until a seat becomes available. If <i>sec</i> is 0 or <i>=sec</i> is omitted, an unlimited number of retries will be used.	Optional
TSK_PROCESS_WAIT= <i>sec</i>	Set the parallel process time-out (default: unlimited). This specifies the number of seconds the system is allowed to wait for a process to run if the maximum number of parallel processes permitted by the license is already running. If <i>sec</i> is 0, this is the same as not specifying the option, and the system will wait as long as needed for the process to run.	Optional
TSK_NET_TIMEOUT= <i>sec</i>	Set the network time-out (default: 4 seconds). This specifies the number of seconds the system is allowed to wait for a connection.	Optional
TSK_NO_ANONYMOUS	Send user name and system name with the license request (for borrowed licenses the user name and system name are always sent, independent of this option). This is useful for statistics and access information of the local license server. This option is required when you use an access file with a local license server. On Windows make sure the Windows environment variable USERNAME is set to the user name. See Section 4.7, Restricting Access to Licenses .	Optional

Speed up network connection

If you know that only TCP port 80, the last one in the list, is open for transparent access on your firewall, you can speed up the connection process by setting the following variable in the `licopt.txt` file:

```
# Connect to port 80
TSK_LICENSE_PORT=80
```

or a full version:

```
TSK_LICENSE_SERVER=lic1.tasking.com:80;lic3.tasking.com:80
```

Example of an options file

```
TSK_LICENSE_KEY_SW160800=1283-cc0a-5e67-8ae3
TSK_LICENSE_SERVER=lic1.tasking.com:8080;lic3.tasking.com:8080
```

TASKING License Management

```
TSK_LICENSE_PROXY_SERVER=proxy.mydomain.com
TSK_LICENSE_PROXY_PORT=8081
TSK_LOG_VERBOSE=2
TSK_LICENSE_WAIT
TSK_NET_TIMEOUT=10
```

2.5.2. License File

If you installed a node-locked client based license, a license file has been created on your computer (`licfile.txt`). This file contains the license(s) of your product. The `#` character marks the start of a comment.

Example of a license file:

```
#
# TASKING License File
#
# Issue date: 2023-10-20 15:35:43
#
# License type: Node-locked client based license|Joinable
#
# Suitable for the computer with computer ID: 5Dzm-L9+Z-WFbO-aMkU-5Dzm-L9+Z-WFbO-aMkU-MDAy-Y2Zm
#
# License key: 1283-cc0a-5e67-8ae3
#
# Product code: SW16080005060000
#
SW16080005060000=71R4BfrOqSwOUF14DTfNsrluhHEbVNaM
O7B/BrtrkJ2KksBYnnB2D2l6TIHtBB9EDGbyi0fDE2je6RmTv
JdlBiiQMFQ792pZT4fU7oArmAtSVx3e6XwyyaSzlVdV97Z81TJ
PS7Pe60V14wESjWG18Q9G5yQfe4QKAzeWexkghdCBDd8RIdq+g
sgA==
```


Chapter 3. Local License Server

This chapter contains an overview of the TASKING local license server and describes how you can install it and use it.

3.1. Overview

Server based licenses can be serviced by either the TASKING remote license server or a local license server. The remote license server is accessed over the Internet whereas a local license server is accessed via your office network or your companies intranet.

A remote license server has the following advantages:

- Ease of use since you do not have to setup a local license server.
- TASKING can generate a license and license key without knowing the Computer ID in advance. So, you can install the package and run it without delay, without sending the Computer ID.
- You can easily update, upgrade (e.g. from Standard to Premium Edition), add extra seats and extend your license.
- The TASKING remote license server has an uptime of 99,9%, through four servers at two different locations.

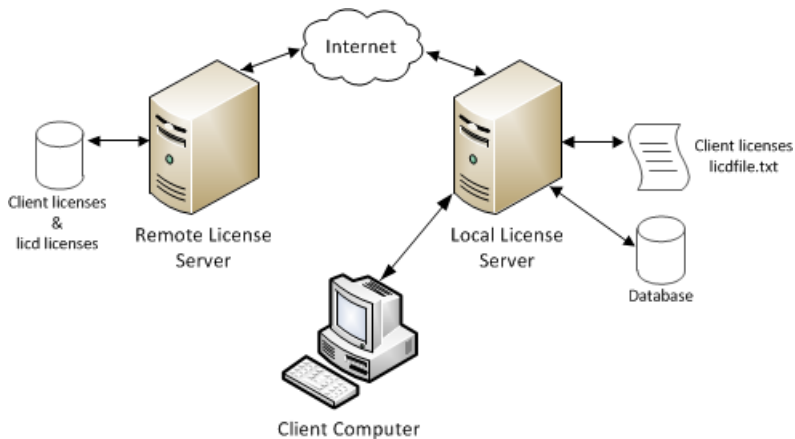
A local license server has the following advantages:

- Enables you to inspect license usage.
- Enables you to manage access to the license server and the licenses.
- Provides facilities for centralized bulk product distributions.
- Provides facilities for license management and transfer of licenses.
- Reduces internet traffic and associated delays.

The following restrictions apply to a local license server:

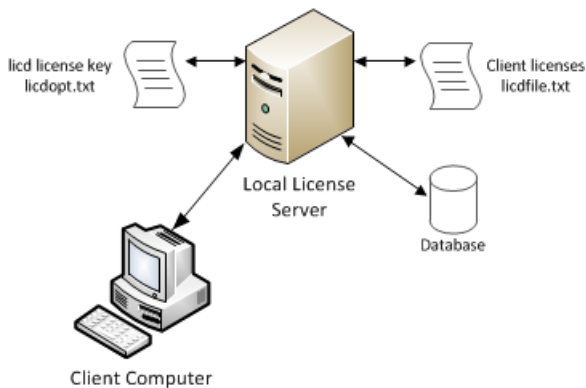
- By default the local license server will not execute on a virtual system. To use a local license server on a virtual system you need a special license. Also your virtual system must be configured to have a static CPU ID and a static MAC address.
- Windows 7 (64-bit), Windows 10 (64-bit) or Windows Server 2008R2 (64-bit) or higher.
- Linux (64-bit) distributions must support GLIBC version 2.17 or higher. You can check the version with the command `ldd --version`

Solution 1: Local license server with internet access to remote license server



In this situation, the local license server licenses itself against a remote license server at first startup. This requires internet access. If internet access is unavailable or not permitted, use solution 2.

Solution 2: Local license server without internet access



In this situation, the local license server licenses itself by reading its license data from a local key. This key must be obtained from TASKING support and added to `licdopt.txt`.

Client licenses

The local license server can retrieve licenses from a license file (`licdfile.txt`) or can retrieve licenses from the remote license server, depending on the chosen solution. The license file is read at startup.

Database

The local license server maintains all licenses and current license usage in a database. The licenses in the database are refreshed from the license file or from a remote server at certain intervals. When you modify the license file `licdfile.txt`, for example, to add a license, you do not need to restart the license server, since the license file is re-read automatically. When the local license server terminates, the current license usage is preserved in the database and is used again at startup. The database is locked to the security ID of the local license server's hardware. Therefore, you cannot move the database to another computer.

3.2. Replacing the Local License Server with a Newer Version

When you want to replace an older version of the local license server with a new one, follow these steps.

1. Make a backup of your `licdfile.txt` file.
2. Install the new version of the local license server as explained in [Section 3.3, *Installing a Local License Server*](#). You do not need to uninstall the older version (if you want to revert back).

If the security ID has not changed, you only need a new license key to activate your new local license server. You can still use the same TASKING product licenses from `licdfile.txt`. If the security ID has changed, you also need to ask for new license keys for the TASKING products serviced by the local license server and you need to update the license keys in `licdfile.txt`.

3. Stop and delete any current running local license server service or daemon or process. See [Section 3.3.4, *Stop and Delete the TASKING License Service on Windows*](#) or [Section 3.3.5, *Stop and Remove the taskinglm Service on Linux*](#).

Note that on Windows you can run `licd --uninstall` from the new installation to stop and delete a running service from a previous version.

4. Start the new local license server. You can start the local license server as a service, daemon or from the command line. See [Section 3.3.1, *Start the Local License Server as a Service on Windows*](#), [Section 3.3.2, *Start the Local License Server as a Daemon on Linux*](#) or [Section 3.3.3, *Start the Local License Server from the Command Line*](#).

Read the rest of this chapter for more information on the local license server. See also [Chapter 4, *Administering Licenses*](#).

3.3. Installing a Local License Server

Install the local license server

1. Download the TASKING Local License Server. See <https://www.tasking.com/support/tlm/downloads> for more information on how to obtain the download link.
2. For Windows follow these steps:
 1. Unpack the ZIP file and run the program **setup.exe**.

TASKING License Management

The TASKING Setup dialog box appears.

2. Click on the **Install** button.
3. Follow the instructions that appear on your screen and **Finish** the installation.
3. For Linux follow these steps:

1. Unpack the ZIP file in a temporary location.

A GNU zipped TAR file is now present (SW000089-vX.YrZ.tar.gz)

2. As root user unpack the `tar.gz` file in the directory where you want to install the local license server, e.g. `/opt`:

```
tar xzvf SW000089-vX.YrZ.tar.gz -C /opt
```

where, `vX.YrZ` is the version number.

The directory `taskinglm` will now be created under `/opt`.

4. Get the security ID of the computer on which you installed the local license server.
 - Open a command prompt (Windows) or a shell (Linux) in the `bin` directory of the installation directory of the TASKING license server, and enter the following command:

```
licd -I
```

The security ID is displayed.

For more information, or when you have problems with a changed security ID because of changed hardware, see [Section 3.3.6, Obtaining a License for a Local License Server](#).

5. Send the security ID to TASKING by email: `licadmin@tasking.com`
6. You will receive an email in return with the license key to activate your local license server, attached as file `licdopt.txt`. On Windows save this file into directory `C:\Tasking` (default) or into a directory of your choice. On Linux save the file for example to `/opt/taskinglm/etc`. Use option **-W** *working-directory* to specify a non-default working directory (see [Section 3.4, Local License Server Options and Defines](#)). This file contains a line in the form:

```
LICD_KEY=abcdefghijklmnopqrstuVwxy2ZgBtWZuKUIiKTrxB98rUPkNdxI6FDlSN  
jR8KLFVCnMM7Axr8fYRPzfDMw724xf93J1q3gFw3jGGTYUCNC3gz4eZVnZw==
```

7. In a second mail, you will receive the 16-character license key that is used for installing the TASKING product on your clients (see [Section 2.4, Installing a License](#)). If you also have received license data for the TASKING product, store it in a file, for example `licdfile.txt`. On Windows put this file into directory `C:\Tasking` (default) or into a directory of your choice. On Linux put the file in the directory `etc` under the installation directory of the local license server, e.g. `/opt/taskinglm/etc`. Use option **-W** *working-directory* to specify a non-default working directory (see [Section 3.4, Local License Server Options and Defines](#)).

If the directory does not exist you need to create the directory yourself.

8. Add the following line to the options file `licdopt.txt`:

Windows:

```
LICD_LICFILE=C:\Tasking\licdfile.txt
```

If you have stored the `licdfile.txt` in another place, replace `C:\Tasking` with the path from step 7.

Linux:

```
LICD_LICFILE=/opt/taskinglm/etc/licdfile.txt
```

If the file `licdopt.txt` does not exist, create one and put it in the same directory as you stored the license file.

You can start the local license server as a service, daemon or from the command line.

3.3.1. Start the Local License Server as a Service on Windows

Install and start the local license server as a service on Windows

1. Open a command prompt as Administrator, navigate to the `bin` directory of the local license server installation, and install the local license server as a service:

```
licd --install [-Wworking-directory] [option]...
```

Note that without administrator rights, you cannot install a service. Also note that you can specify other command line options or set defines in `licdopt.txt`. When you start the local license server as a service, this file must be present in the directory `C:\Tasking` or in the directory specified with option `-Wworking-directory`. See [Section 3.4, Local License Server Options and Defines](#) for a list of options and defines.

2. Open a command prompt as Administrator and start the local license server as a service:

```
net start "TASKING License Service"
```

Note that without administrator rights, you cannot start a service.

3.3.2. Start the Local License Server as a Daemon on Linux

Install and start the local license server as a systemd service on Linux

1. Open a shell prompt and as root set up a systemd service to start it by creating the file `/etc/systemd/system/taskinglm.service` with the following contents (assuming the license manager is installed in `/opt/taskinglm`):

```
[Unit]
Description=TASKING License Manager
After=syslog.target network.target
```

TASKING License Management

```
[Service]
Type=forking
ExecStart=/opt/taskinglm/bin/licd
```

2. Open a shell prompt and as root start the local license server as a service:

```
systemctl daemon-reload
systemctl start taskinglm
```

3.3.3. Start the Local License Server from the Command Line

The preferred way is to start the local license server as a service, but when necessary, for example for debugging purposes or on Linux, you can start the local license server from the command line.

1. Windows only: Open a command prompt as Administrator.
2. Start the local license server:

```
licd [option]...
```

Instead of specifying options on the command line, you can set defines in `licdopt.txt`. This file must be present in the current working directory or in the `etc` or `bin` directory relative to the current working directory or the directory specified with option `-Wworking-directory`. See [Section 3.4, Local License Server Options and Defines](#) for a list of options and defines.

3. (Optional) When you get a security alert from a firewall, allow access to the network(s) that **licd** needs access to.

You can only run one instance of the license server on a system. If you start a second one, the first one stops. This will lead to unexpected results.

3.3.4. Stop and Delete the TASKING License Service on Windows

How to stop and delete the TASKING License Service on Windows

- Open a command prompt as Administrator, navigate to the `bin` directory of the local license server installation, and stop and delete the local license server as a service:

```
licd --uninstall
```

Note that without administrator rights, you cannot uninstall a service.

Also note that you can use this command to stop and delete a previous version of the local license server as a service.

3.3.5. Stop and Remove the taskinglm Service on Linux

How to stop and remove the taskinglm service on Linux

If you want to uninstall the local license server, you must first stop and remove the service.

- Open a shell prompt and as root stop and remove the local license server as a service:

```
systemctl stop taskinglm
systemctl disable taskinglm
rm /etc/systemd/system/taskinglm.service
systemctl daemon-reload
systemctl reset-failed
```

3.3.6. Obtaining a License for a Local License Server

A local license server is licensed with the security ID of the computer on which you installed the local license server. When the local license server starts it contacts the remote license server with its security ID to get its license, or if you received a license key in `licdopt.txt` it verifies its security ID against this key locally.

The security ID of a local license server is unique. You can obtain the security ID with the following command:

```
licd -I
```

On first need for an security ID the system takes a snapshot of the hardware. On consecutive needs for a security ID the hardware will be compared with the snapshot. If you changed anything to your hardware configuration and the deviation is within limits, the original security ID will be used and your license key is still valid. If the deviation is outside limits there is a security ID mismatch, meaning that your license key is no longer valid. If you use the `-I` option when the deviation is outside limits, the local license server will report:

```
System deviates too much from its Security ID. Restore your previous
hardware configuration, or combine option -I with -F to update the
Security ID to match the current hardware. If the Security ID has
changed, please request a new local license server key from TASKING.
Note that all product licenses served from a license file need to be
updated by TASKING as well.
```

Only when you run `'licd -I -F'` the snapshot will be updated and the security ID will change. This invalidates the emailed local license server key and you will need to send the new security ID to TASKING to get a new local license server key. Also you need to request new license keys for all product licenses served from a license file by the local license server. Therefore, only use this option combination when instructed to do so, because when the system deviates a little and you still use `'-I -F'`, the snapshot will be updated and the security ID will change.

If you install a new version of the local license server, this invalidates the database. However, the security ID is still the same, so the product licenses serviced by the local license server are still valid. In this case you only need a new local license server key. After startup the local license server validates itself at the

TASKING License Management

remote license server, or with the local license key, and retrieves the client data on demand from a license file or the remote license server and builds a new database.

3.4. Local License Server Options and Defines

You can set defines in the file `licdopt.txt` to control the local license server. The defines in `licdopt.txt` are read by the local license server when installed as a service or when started from the command line. If `licdopt.txt` is present, this file is always read.

When you start the local license server from the command line, the file `licdopt.txt` is searched for in the following path:

Windows:

```
..\etc;<licddir>\..\etc;<licddir>\..\bin;<licddir>
```

Linux:

```
../etc;<licddir>/../etc;<licddir>/../bin;<licddir>
```

where `.` represents the current working directory and `<licddir>` is the directory where **licd** is installed.

When you start the local license server as a Windows service, the `.` is replaced by `\Tasking` in the above search path.

When you start the local license server with option **-W** *working-directory*, the `.` or `\Tasking` is replaced by *working-directory* in the search path. This must be an absolute path and can include a drive letter specification under Windows.

Any `#` character in the `licdopt.txt` file is considered as the start of a comment.

The option flags mentioned in the table below are available on the command line only. Command line options have precedence over defines from the options file. Note that you cannot use environment variables to set options for the local license server. The following options and defines are available:

Option	Define in <code>licdopt.txt</code>	Description
<code>-?</code>		Show option summary
<code>-A[access-file]</code>	<code>LICD_ACCESSFILE=file</code>	Use an access file with access rules (default for <code>-A</code> : <code>licdaccess.txt</code>). See Section 4.7, Restricting Access to Licenses .
<code>-D</code>		Display the values of the used options and defines, the option file search path and which option file is loaded. This can be useful for debugging licensing problems.
<code>-E[event-file]</code>	<code>LICD_EVENTFILE=file</code>	Log license management event information in a file (default for <code>-E</code> : <code>licd.csv</code>). See Section 3.7, License Management Event Log .

Option	Define in licdopt.txt	Description
-f [<i>license-file</i>]	LICD_LICFILE= <i>file</i>	Use a license file (default for -f : <code>licdfile.txt</code>). The search path is the same as with the options file specified above this table. This option is mutually exclusive with -r / <code>LICD_USE_REMOTE</code> .
-h <i>hostname</i>	LICD_REM_SERVER= <i>host</i> [: <i>port</i>] [...]	Use the remote license server for LicD validation and if requested, client licenses (default: <code>lic3.tasking.com:8080</code>). Possible ports you can use are 8080, 8936 or 80.
-I		Show Security ID of the computer that executes the local license server.
-L [<i>logfile</i>]	LICD_LOGFILE= <i>file</i>	Log information in a file (default for -L : <code>licd.log</code>).
-l		Use the syslog (Linux only).
-n		Run LicD, but not as a daemon (Linux only).
-P <i>port</i>	LICD_REM_SERVER_PORT= <i>port</i>	Connect to port <i>port</i> on the remote license server (default: 8080). Possible ports you can use are 8080, 8936 or 80.
-p <i>port</i>	LICD_LISTEN_PORT= <i>port</i>	Listen on port <i>port</i> for communication with clients (default: 9090).
-Q <i>port</i>	LICD_PROXY_PORT= <i>port</i>	Specify the TCP port number to use for a proxy server (default: 3128).
-q <i>proxyserver</i>	LICD_PROXY_SERVER= <i>proxyserver</i> [: <i>port</i>]	If a proxy server is required for outgoing license requests, use this option to specify <i>proxyserver</i> as the HTTP proxy server. If you omit the <i>port</i> , the default port is 3128.
-R <i>num</i>	LICD_AUTOREFRESH_TIMER= <i>num</i>	Auto-refresh timer. When the auto-refresh timer expires, the license server retrieves all licenses from either the license file (-f / <code>LICD_LICFILE</code>) or from the remote license server (-r / <code>LICD_USE_REMOTE</code>). The number <i>num</i> specifies the refresh time in seconds. Valid values are between 60 and 14400 seconds (default: 600 seconds).
-r	LICD_USE_REMOTE	Use the remote license server to get client licenses. This option is mutually exclusive with -f / <code>LICD_LICFILE</code> .
-v		Display version header only (the tool exits with exit code 1).
-v	LICD_LOG_VERBOSE	Verbose, repeat for more verbose.
	LICD_KEY= <i>key_data</i>	If network connectivity is unavailable, a local license server can read its license data from a local license key. This key must be obtained from TASKING support by providing the security ID with <code>licd -I</code> .

TASKING License Management

Option	Define in licdopt.txt	Description
	LICD_MASTER= <i>host</i> [: <i>port</i>]	In a redundant master - slave configuration, use this option to specify the master server in the option file on the slave server. See Section 3.5, Redundant Master - Slave Configuration .
	LICD_QUEUE_MODE=ON OFF	Globally turn license queueing on or off. See Section 4.5, Using a License Queue .
	LICD_REFUSE_ANONYMOUS	Do not allow anonymous access to the local license server. This option forces that every client publishes his user name and system name with the client option TSK_NO_ANONYMOUS.
<code>-working-directory</code>		Use <i>working-directory</i> instead of the default current directory or <code>\Tasking</code> if running as a Windows Service. The <i>working-directory</i> must contain an absolute path. Under Windows the <i>working-directory</i> can contain a drive letter specification.
<code>--install</code>		Install the local license server as a Windows Service (Windows only, run as Administrator).
<code>--uninstall</code>		Uninstall the local license server as a Windows Service (Windows only, run as Administrator).

Example of a local license server options file:

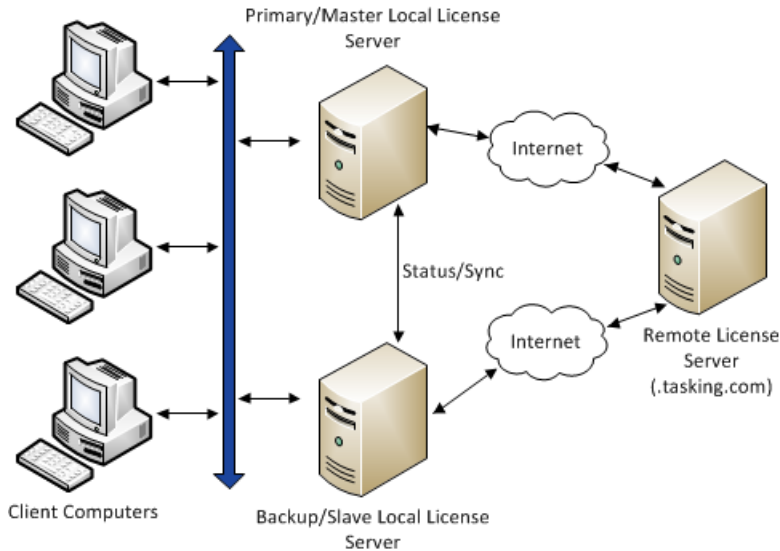
```
LICD_LICFILE=licdfile.txt
LICD_LOGFILE=mylog.log
LICD_EVENTFILE=licdevents.csv
LICD_ACCESSFILE=myveryspecialaccess.txt
LICD_LOG_VERBOSE
LICD_QUEUE_MODE=ON
LICD_REM_SERVER=lic3.tasking.com:8080
LICD_LISTEN_PORT=9090
LICD_PROXY_SERVER=proxy.mydomain.com
LICD_PROXY_PORT=8081
LICD_AUTOREFRESH_TIMER=300
```

3.5. Redundant Master - Slave Configuration

In a redundant server configuration there is a master local license server and a slave local license server. In normal operation mode the master server serves the licenses for the clients and the 'Sleeping Slave' server synchronizes its own database every minute with the master database. The slave server refuses to serve licenses.

If the master becomes unreachable, the slave server changes its operation mode to 'Active Slave'. It tries to get a time limited slave license from a TASKING remote license server and starts serving licenses with the last known master status. If the master server is back online again, the slave server changes its operation mode to 'Sleeping Slave' and starts the sync again with its master. The status of the slave server is not synchronized back with the master.

A slave server cannot read license files and cannot get license data from a TASKING remote license server except its own slave license. It can only get data from its master.



Client configuration

To use a redundant master - slave configuration, the client options file `licopt.txt` must contain both servers. For example,

```
TSK_LICENSE_SERVER=master.domain.com:9090;slave.domain.com:9090
```

If the master becomes unreachable a client switches to the next server in the list, the slave. If the master comes online again the slave refuses license requests. The client switches back to the first server in the list, the master.

A redundant local license server configuration needs two local license server licenses, one for the master and one for the slave (a time limited slave license).

The master and slave license servers must have the same version number and must run on the same operating system.

Slave server configuration

The file `licdopt.txt` of a slave server must contain the following configuration item:

```
LICD_MASTER=hostname[:port]
```

For example:

```
LICD_MASTER=lic-master.domain.com:9090
```

3.6. Creating a Local License Server License File

When you receive a license that must be serviced by a local license server then the email that you receive contains a license key and the license data. You have to copy the license data to the license file (default: licdfile.txt).

Example of a license email:

Dear user,

Please find below the license for the 07-200-160-808
(TASKING VX-toolset for TriCore, Premium Edition) version 6.0.

The license key for your floating server license for 10 seats is:

e3e9-daef-d1c9-aa91

Software Product code : SW16080005060
License serial number : 12345678
Platform : Windows

Serviced by the license server with Security ID: hg1-KvZ4-Q70W-qBoG

Below you'll find details to get started:

License Installation

If you haven't done so already, please download your TASKING toolset [here](#).

Once downloaded, follow the steps in our [License Options Page](#) to install your software and license.

Need Support?

- [Support Home](#) - Access documentation and downloads for your TASKING toolset.
- [Contact Us](#) - Get in touch with a TASKING support representative.

Best regards,

Your TASKING Team

- Copy the following data to your license file licdfile.txt, default in: C:\Tasking, on the system that will be running the license server -

```
SW16080005060000=4879pXXTSdt6WNMg5YDnkJvxLxFIP+Cqz  
/azB84oPIyFgJrIXymgVdlqM8odSKkFSL04Vdu7IissrsLVJGQ  
yw6qDAPK3beos7Kc62UCICJ1SL3oCIFw7QT7CvD7tSNnV6Wsq0  
9e380BHqOhtzVNxAjku8R+X7qLwqhttdZ2HhpRnOtN76WY1BT2
```

```
4BcDSQoen5N+A5QmY6BrqCrbKwNJpseOGNR5MiXLe+Ijk635md
V/9L0j/vufm0Uvc9XzQfPaTpxqoOR80Vhbo06UHKQ==
```

Note that when you have multiple license key entries in `licdfile.txt` you must separate the keys with at least one empty line, or comment line. For example:

```
SW16080005060000=4879pXXTSdt6WNNg5YDnkJvxLxFIP+Cqz
/aZB84oPIyFgJrIXymgVd1qM8odSKkFSL04Vdu7IissrsLVJGQ
yw6qDAPK3beos7Kc62UCICJ1SL3oCIFw7QT7CvD7tSNnV6Wsq0
9e380BHqOhtzVNxAjku8R+X7qLwqhttdZ2HhpRnOtN76WY1BT2
4BcDSQoen5N+A5QmY6BrqCrbKwNJpseOGNR5MiXLe+Ijk635md
V/9L0j/vufm0Uvc9XzQfPaTpxqoOR80Vhbo06UHKQ==
```

```
SW16080005050000=5wZBpxCwRRucQgRQNipXuSdptwSPgeZxb
twndcFx558Bztq9Ct4rCo/XxqghDr200FrZYPjeg9T2NTOxYV+
xSCTejMVSS+9WUDFzu+JzGIBELbZotvNiCL0118bBWr4vAovsp
8ZSi jdCZNXZrCkj2xK5N1VqEDa21L3cJ71vHPHv9817znvA9ma
Qy8JvXw2fb0cHgU1nEs6gLwtleoQ4gEmVMeoMmh0z14BqpR/Wc
gi6Wu0T9j9YU8S7WpHZjA1UFapL6etbkCMBWSMnRw==
```

If a duplicate license key is found, a warning is given and the last duplicate in the file is taken.

3.7. License Management Event Log

With command line option `-E[event-file]` or define `LICD_EVENTFILE=file` in `licdopt.txt`, the TASKING local license server LicD can log incoming license requests and their results in a license management event log.

The event log file is a Comma Separated Values (CSV) file with a single header defining the fields and one or more detail records of the event. The log file includes the status, including the number of used license seats, after the request.

The following fields are present:

Field	Description
Date	Date part of the timestamp of the record <code>YYYY-MM-DD</code> .
Time	Time part of the timestamp of the record <code>HH:MM:SS</code> .

TASKING License Management

Field	Description
Event	<p>Possible events are:</p> <ul style="list-style-type: none"> • Granted The request was granted, license count was incremented. • Extended The request was granted while re-using the seat of the previous request, license count was not changed. This is equivalent to a <code>Freed</code> and subsequent <code>Granted</code> event. • Denied The request was denied, license count was not changed. • Freed The seat was freed (by request of clean-up process), license count was decremented.
User	Username (or empty if unknown).
System	Hostname (or empty if unknown).
IP	IP address.
Request	<p>Possible values are:</p> <ul style="list-style-type: none"> • The license key <code>xxxx-xxxx-xxxx-xxxx</code>. • The full Software Code in case of a joinable license <code>SWpppppphhvvv000</code>. • The pool name <code>POOL-xxxxxxxxxxxxxxxx</code>.
SW-code	The Software Code <code>SWpppppphhvvv000</code> .
Type	floating or node-locked.
Key	Actual license key <code>xxxx-xxxx-xxxx-xxxx</code> .
Seats	Total number of seats available for this 'Key'.
Used	Total number of used/active seats for this 'Key'.
Seat-ID	Seat identifier in the form <code>xxxxxxxx</code> (equals Request-ID of a similar initial request).
Request-ID	Request identifier of the current request in the form <code>xxxxxxxx</code> .
Duration	Duration in seconds of validity for the requested license (linger time for a regular request, borrowing time for a borrow request).
Info	Extra textual information on this record. Usually this contains LicD log information.

The following example shows parts of an event log:

Date;Time;Event;User;System;IP;Request;SW-code;Type;Key;Seats;Used;Seat-ID;Request-ID;Duration;Info

- 1 2023-01-31;16:31:29;Granted;;;127.0.0.1;1111-1111-1111-1111;SW16080005063000;floating;1111-1111-1111-1111;2;1;047dd2a5;047dd2a5;120;...
- 2 2023-01-31;16:31:29;Extended;;;127.0.0.1;1111-1111-1111-1111;SW16080005063000;floating;1111-1111-1111-1111;2;1;047dd2a5;0fbc3c86;120;...
- 3 2023-01-31;16:42:21;Denied;;;127.0.0.1;2222-2222-2222-2222;SW16080002063000;unknown;;0;0;9e52fb31;9e52fb31;0;...
- 4 2023-01-31;16:44:51;Granted;user;system;127.0.0.1;SW16080002063000000;SW16080002063000;floating;2222-2222-2222-2222;1;1;386fbe90;386fbe90;120;...
- 5 2023-01-31;16:44:51;Granted;user;system;127.0.0.1;SW16080002063000000;SW16080002063000;floating;3333-3333-3333-3333;1;1;d50a1e6c;d50a1e6c;86400;...
- 6 2023-01-31;16:44:51;Freed;user;system;127.0.0.1;SW16080002063000000;SW16080002063000;floating;2222-2222-2222-2222;1;0;386fbe90;d50a1e6c;86400;...

On line 1 an initial license request was Granted and the Seat-ID gets the same value as the initial Request-ID.

On line 2 the license of line 1 was Extended. The Seat-ID stays the same as the initial one (from line 1) and the Request-ID gets a new value because of the new request.

On line 3 a license request was Denied because, for example, there were no more seats available.

On lines 4, 5 and 6 the following has happened: On line 4 a license request from a license pool was Granted. A borrowing request came in, but the license from line 4 was not marked as being borrowable. Line 5 shows that another license from the pool could be borrowed and this was Granted and on line 6 the seat from line 4 was Freed.

TASKING License Management

Chapter 4. Administering Licenses

This chapter contains information about handling and monitoring licenses. You can monitor the status of the local license server, borrow licenses, use licenses in a license pool and license administrators can restrict access to licenses.

4.1. Monitoring the Status of the Local License Server

Browser

You can inspect the actual status and see the log information of the local license server with a browser. Use the following address:

```
http://lic_server_hostname:9090/stats
```

where, *lic_server_hostname* is the IP address or DNS name of the local license server.

For example the status page can look similar to this:

TASKING License Server vx.yrz Status - Master

Available Product Licenses

License key	Serial#	Product code	Valid from	Expires	Expiration type	Modified	Type	Seats	In use	Queued	Processes	License pools
001 aaaa-aaaa-aaaa-aaaa	01234567	SW16080005063000	2021-07-30	-	-	2025-02-27 11:03:13	Float	2	2	1	-	POOL-tric-stand-dard
002 bbbb-bbbb-bbbb-bbbb	12345678	SW16080002063000	2021-10-05	-	-	2025-02-27 11:02:53	Float	2	0	-	-	POOL-tric-stand-dard
003 cccc-cccc-cccc-cccc	23456789	SW16080002063000	2023-09-28	-	-	2025-02-27 11:03:24	Float	2	1	-	-	POOL-tric-stand-dard

License Seats In Use

License key	Serial#	Product code	Queued since	In use since	Refreshed	Borrowed until	User	System
001 aaaa-aaaa-aaaa-aaaa	01234567	SW16080005063000	-	2025-02-27 11:03:13	-	2025-02-28 11:03:13	first.lastname	PCNAME1
			2025-02-27 11:03:31	2025-02-27 11:03:30	2025-02-27 11:03:30	-	First2	Last2 PCNAME2
003 cccc-cccc-cccc-cccc	23456789	SW16080002063000	-	2025-02-27 11:03:24	2025-02-27 11:03:24	-	user unknown	123.45.6.78
							user unknown	123.45.6.79

Logging

```
2025-02-27 11:03:33 Autosave
2025-02-27 11:03:31 [87c0d38b] No free seat for floating license key aaaa-aaaa-aaaa-aaaa
2025-02-27 11:03:31 [87c0d38b] Queued for aaaa-aaaa-aaaa-aaaa - SW16080005063000 at position 1 of 1, 0 free seats available
2025-02-27 11:03:31 [87c0d38b] Connect from 123.45.6.78 (/) key aaaa-aaaa-aaaa-aaaa - SW16080005063000
2025-02-27 11:03:30 [b24a0aaa] Valid floating license aaaa-aaaa-aaaa-aaaa - SW16080005063000 [b24a0aaa]
2025-02-27 11:03:30 [b24a0aaa] Connect from 10.200.123.457 (First2 Last2/PCNAME2) key aaaa-aaaa-aaaa-aaaa - SW16080005063000
2025-02-27 11:03:24 [7ec6097c] Valid floating license cccc-cccc-cccc-cccc - SW16080002063000 [7ec6097c]
2025-02-27 11:03:24 [7ec6097c] Connect from 123.45.6.79 (/) key cccc-cccc-cccc-cccc - SW16080002063000
2025-02-27 11:03:13 [de2855a8] License borrowed for 86400 seconds until: 2025-02-28 11:03:13
2025-02-27 11:03:13 [de2855a8] Valid floating license aaaa-aaaa-aaaa-aaaa - SW16080005063000 [de2855a8]
2025-02-27 11:03:13 [de2855a8] Connect from 10.200.123.456 (first.lastname/PCNAME1) key aaaa-aaaa-aaaa-aaaa - SW16080005063000
2025-02-27 11:03:13 Fixup
2025-02-27 11:03:03 Autosave
2025-02-27 11:02:53 Reading license file done
2025-02-27 11:02:53 Adding license cccc-cccc-cccc-cccc from line 51 to POOL-tric-stand-dard (SW16080002063000 serial=23456789 seats=2)
2025-02-27 11:02:53 Adding license bbbb-bbbb-bbbb-bbbb from line 36 to POOL-tric-stand-dard (SW16080002063000 serial=12345678 seats=2)
2025-02-27 11:02:53 Adding license aaaa-aaaa-aaaa-aaaa from line 9 to POOL-tric-stand-dard (SW16080005063000 serial=01234567 seats=2)
2025-02-27 11:02:53 Reading license file licdfile.txt
2025-02-27 11:02:53 Opening licdfile.txt
2025-02-27 11:02:53 Reading options file ./licdopt.txt
2025-02-27 11:02:53 Opening ./licdopt.txt
2025-02-27 11:02:53 Searching for licdopt.txt in ./etc;/opt/taskinglm_vx.yrz/bin/./etc;/opt/taskinglm_vx.yrz/bin/./bin;/opt/taskinglm_vx.yrz/bin/
```

The first table shows the actual status of the available product licenses. The second table, if present, shows information about the license seats in use. Below the table you can find logging information of the local license server.

TASKING License Management

The following columns are available in the **Available Product Licenses** table:

Column	Description
ID	Internal ID of the license.
License key	The license key.
Serial#	The serial number of the product.
Product code	The product code.
Valid from	The date at which the license becomes valid.
Expires	The expiration date of the license, or '-' when no expiration date is set.
Expiration type	The type of expiration date: Maintenance (software maintenance expiration date), License (license expiration date), - (no expiration date set).
Modified	The date the license was modified.
Type	The type of license: Float (floating license), Node (node-locked license).
Seats	The number of license seats for a floating license.
In use	The number of license seats currently in use.
Queued	The number of license requests currently waiting in the queue for a free seat.
Processes	The maximum number of concurrent processes per seat.
License pools	The pools a license belongs to.

The following columns are available in the **License Seats In Use** table:

Column	Description
ID	Internal ID of the license.
License key	The license key.
Serial#	The serial number of the product.
Product code	The product code.
Queued since	The date and time since the license request is in the queue.
In use since	The date and time since the license seat is in use.
Refreshed	The date and time the license seat is last refreshed (regular license only).
Borrowed until	The end date and time until the license seat is borrowed (borrowed license only).
User	The user name (if available). For borrowed licenses, the user name is always shown, independent of the setting of <code>TSK_NO_ANONYMOUS</code> .
System	The system name (if available) or the IP address. For borrowed licenses, the system name is always shown, independent of the setting of <code>TSK_NO_ANONYMOUS</code> .

If you want to log the information in a file, you can use option **-L** of the local license server:

```
licd -L licd.log
```

Command line

Instead of using a browser you can also use the tool **licdstat** to see the status and log information of a local license server.

```
licdstat [option]...
```

The following options are available:

- ?** show option summary
- afile** append output to a file, mutually exclusive with **-f**
- ffile** write output to a file, mutually exclusive with **-a**
- hhost** host to query (default: localhost)
- key** query a specific key
- l** list current license status
- pport** TCP port to use (default: 9090)
- t** display server status and version
- u** list current users
- V** display version header only (the tool exits with exit code 1)
- v** verbose output
- X** generate X-Formation compatible output

Note that the user name and system name in the users lists of a license is only available if the option `TSK_NO_ANONYMOUS` is present in `licopt.txt` on the respective client computers. On Windows also make sure the Windows environment variable `USERNAME` is set to the user name.

Example:

```
licdstat -l -u -t
```

```
LicD: vx.yrz localhost:9090 Thu 27 Feb 2025 09:24:54 CET SecurityID: 3cHE-GME0-FW3I-sHTW
```

```
08f3-0d7b-9fe9-25de SW16080002 v5.0 (floating ser#:99224421 seats:10 inuse:10 borrowed:0
                                queued:- processes:16)
```

```
Firstname1 Lastname1/pc_a      (v4.9 2025-02-27 09:24:11)
Firstname2 Lastname2/pc_b      (v5.0 2025-02-27 09:24:11)
Firstname3 Lastname3/pc_c      (v4.1 2025-02-27 09:24:12)
Firstname4 Lastname4/pc_d      (v4.6 2025-02-27 09:24:12)
Firstname5 Lastname5/pc_e      (v4.9 2025-02-27 09:24:13)
Firstname6 Lastname6/pc_f      (v4.4 2025-02-27 09:24:13)
Firstname7 Lastname7/pc_g      (v4.4 2025-02-27 09:24:14)
Firstname8 Lastname8/pc_h      (v4.4 2025-02-27 09:24:14)
Firstname9 Lastname9/pc_i      (v4.1 2025-02-27 09:24:15)
Firstname10 Lastname10/pc_j     (v5.0 2025-02-27 09:24:15)
```

TASKING License Management

```
5854-61b9-4ecb-04f0 SW16080002 v6.0 (floating ser#:99015724 seats:2 inuse:2 borrowed:0
                                                                    queued:1 processes:8)

Firstname11 Lastname11/pc_k      (v4.4  2025-02-27 09:24:16)
Firstname12 Lastname12/pc_l      (v4.1  2025-02-27 09:24:16)
Firstname13 Lastname13/pc_m      (v6.0  2025-02-27 09:24:45) queued until 2025-02-27 09:24:45
```

You can use **licdstat** to generate output for the license statistics and management system X-Formation. The output is an XML status file which can be read by X-Formation. The following example is a Windows batch file that updates the license statistics for X-Formation:

```
:loop
  licdstat -h tlmserver.example.com -X -f c:\tlmlogs\tlm.xml
  timeout 60
  goto :loop
```

Instead of a batch file you can use the Windows Task Scheduler to call **licdstat** on a regular basis.

Exit code

The **licdstat** tool has the following exit codes:

0	Success
1	Error or version header output

4.2. Transferring a License (Borrow Feature)

It is possible to move a floating license to a node-locked client based license temporarily. This is called 'borrowing'. You can use the tool **licborrow** to obtain a time restricted node-locked client based license on a client system from a local license server servicing a floating license for that client.

The license is retrieved by specifying the product code of the TASKING product. You can find the product code in the email or paper that contains the license key. The product code consists of the letters SW and the first 6 digits of the SW code, for example SW160800.

You can restrict access to which users and systems are allowed to borrow a license. A 0 days request returns the borrowed license to the TASKING server.

To borrow a license (Windows, GUI)

1. From the **Start** menu, select **Programs » TASKING product vx.yrz » Borrow License**.

The Borrow License dialog appears.

2. From the **Borrow license for** list, select the product for which you want to borrow a license.
3. Enter the number of **Days** you want to borrow the license. One day literally means 24 hours.
4. Click **Borrow License**.

To borrow a license (Windows, command line)

On Windows a command line version is available also. You can download it from <https://www.tasking.com/support/tlm/downloads>. Enter the following command at a Windows command prompt:

```
licborrowcmd [option]... product-code version
```

where,

<i>product-code</i>	a valid software product code in the form SWxxxxxx, where xxxxxx is a 6 digit number.
<i>version</i>	a valid version in the form vX.YrZ, where X is the major version, Y is the minor version and Z is the revision number. For example, v6.3r1. If the rZ is omitted, r1 is assumed, if .YrZ is omitted vX.0r1 is assumed.

The following options are available:

-?	show option summary
-d <i>days</i>	number of days to borrow (0 .. 128), 0 returns a license
-L	extended list installed products
-l	list installed products
-V	display version header only (the tool exits with exit code 1)
-v	verbose

Examples

Show a list of installed products:

```
licborrowcmd -l
```

```
SW160800 TASKING VX-toolset for TriCore v6.0r1
SW160800 TASKING VX-toolset for TriCore v6.1r1
SW160800 TASKING VX-toolset for TriCore v6.2r2
SW160800 TASKING VX-toolset for TriCore v6.3r1
```

Borrow a license for 7 days:

```
licborrowcmd -d7 SW160800 v6.3r1
```

Return a license, and verbose enabled:

```
licborrowcmd -v -d0 SW160800 v6.3r1
```

```
Getting license info from server for license-key
Connecting to licl.tasking.com:8080
Putting license info into cache for license-key
License for SW160800 v6.3r1 successfully returned to licl.tasking.com
```

TASKING License Management

Show license details about the installed products:

```
licborrowcmd -L
```

```
SW160800 TASKING VX-toolset for TriCore v6.0r1, cached or borrowed license
with key license-key expired on this system at 2025-02-17 17:04
SW160800 TASKING VX-toolset for TriCore v6.1r1, no cached or borrowed license
with key license-key available on this system
SW160800 TASKING VX-toolset for TriCore v6.2r2, cached or borrowed license
with key license-key expires on this system at 2025-02-24 15:26
SW160800 TASKING VX-toolset for TriCore v6.3r1, cached or borrowed license
with key license-key expires on this system at 2025-02-24 15:26
```

To borrow a license (Linux)

On Linux PCs, enter the following command at a shell command prompt, assuming that the `bin` directory of the TASKING product is in the search path:

```
licborrow [option]... product-code version
```

where,

<i>product-code</i>	a valid software product code in the form SWxxxxxx, where xxxxxx is a 6 digit number.
<i>version</i>	a valid version in the form vX.YrZ, where X is the major version, Y is the minor version and Z is the revision number. For example, v6.3r1. If the rZ is omitted, r1 is assumed, if .YrZ is omitted vX.0r1 is assumed.

The following options are available:

-?	show option summary
-d <i>days</i>	number of days to borrow (0 .. 128), 0 returns a license
-V	display version header only (the tool exits with exit code 1)
-v	verbose

Example

Borrow a license for 7 days:

```
licborrow -d7 SW160800 v6.3r1
```

Return a license:

```
licborrow -d0 SW160800 v6.3r1
```

Exit code

The `licborrow/licborrowcmd` tool has the following exit codes:

0	Success (e.g. borrow was successful)
1	Syntax error or other error (e.g. borrow failed), or version header output

4.3. Joining or Grouping Licenses by Using a License Pool

When you are using a local license server, you can join licenses for a product or group licenses for different products by using a license pool. License pools are defined in the file `licdfile.txt` which is read by the local license server. You can then use a license pool in your license options file `licopt.txt`.

Joining licenses

You can bring together all licenses for one product into one overall license, this is called 'joining licenses'. Suppose you have placed two orders for a given product. In the first order you bought 5 floating licenses, in the second order you bought 10 additional floating licenses. With each order you have received a license key. You now have 15 floating licenses and 2 license keys. One key associated with 5 floating licenses, the other associated with 10 licenses.

If you run six instances of the product that use the license key obtained with the first order then one product instance will fail to execute due to the fact that there are only five licenses available. You do not want such behavior since you have bought 15 licenses. To solve this you can join licenses by using a license pool.

Grouping licenses

To specify that the licenses for one or more products shall be grouped together you can use a license pool. This can be the case for example when your license agreement has constraints on geographical location, such as a single site license, country license or continental license. In that case you can put all product licenses that are allowed to be used by a specific location into a license pool.

For global licenses you can put licenses into multiple license pools by specifying multiple pool names on a single line.

Pool names

For TASKING products built with TASKING License Management v1.5r1 or newer (see the release note of the TASKING product), pool names have the following format:

`POOL-name`

The *name* part is free to choose from 1 up to 14 characters (no whitespace characters). You can use lowercase and uppercase alphanumerical characters.

TASKING License Management

Pool names are also supported with existing TASKING products that have been built with an older version of TLM, as long as you use the new local license server and specify the pool names in the fixed format and length:

```
POOL-xxxx-xxxx-xxxx
```

Only the xxx parts are free to choose as long as all positions are filled in (no whitespace characters). You can use lowercase and uppercase alphanumerical characters.

Note that if you need to continue to use older versions of TASKING products which support the fixed pool name format and length together with newer versions which do support the more flexible format, you need to use a pool name with the fixed format and length.

Defining license pools in licdfile.txt

You can define license pools in the file `licdfile.txt` using the following format:

```
[licenses outside any pool]
```

```
POOL-aaaa-aaaa-aaaa  
licenses for pool POOL-aaaa-aaaa-aaaa
```

```
POOL-bbbb-bbbb-bbbb  
licenses for pool POOL-bbbb-bbbb-bbbb
```

```
POOL-aaaa-aaaa-aaaa POOL-bbbb-bbbb-bbbb  
global licenses for both pools
```

```
POOL-  
[licenses outside any pool]
```

For example when joining licenses:

```
POOL-tric-stan-dard  
SW16080005063000=4879pXXTSdt6WNMg5YDnkJvxLxFIP+Cqz  
/aZB84oPIyFgJrIXymgVdlqM8odSKkFSL04Vdu7IissrsLVJGQ  
yw6qDAPK3beos7Kc62UCICJ1SL3oCIFw7QT7CvD7tSNnV6Wsq0  
9e380BHqOhtzVNxAjku8R+X7qLwqhttdZ2HhpRnOtN76WY1BT2  
4BcDSQoen5N+A5QmY6BrqCrbKwNJpseOGR5MiXLe+Ijk635md  
V/9L0j/vufm0Uvc9XzQfPaTpxqoOR80Vhbo06UHKQ==
```

```
SW16080005063000=5wZBpxCwRRucQgRQNipXuSdptwSPgeZxb  
twndcFx558Bztq9Ct4rCo/XxqqhDr200FrZYPjeg9T2NTOxYV+  
xSCTejMVVs+9WUDFzu+JzGIBEIbZotvNiCL0118bBWr4vAovsp  
8ZSijdCZNXZrCkj2xK5N1VqEDA21L3cJ71vHPHv9817znvA9ma  
Qy8JvXw2fb0cHgUlnEs6gLwtleoQ4gEmVMeoMmh0z14BqpR/Wc  
gi6Wu0T9j9YU8S7WpHZjA1UFapL6etbkCMBWSMnRw==
```


For example when grouping by location:

```
POOL-EURO-PE__-____
# put your country licenses for Europe here
SW16080005063000=4879pXXTSdt6WNMg5YDnkJvxLxFIP+Cqz
/aZB84oPIyFgJrIXymgVdlqM8odSKkFSLO4Vdu7IissrsLVJGQ
yw6qDAPK3beos7Kc62UCICJ1SL3oCIFw7QT7CvD7tSNnV6Wsq0
9e380BHqOhtzVNxAjku8R+X7qLwghttdZ2HhpRnOtN76WY1BT2
4BcDSQoen5N+A5QmY6BrqCrbKwNJpseOGNR5MiXLe+Ijk635md
V/9L0j/vufm0Uvc9XzQfPaTpxqoOR80Vhbo06UHKQ==
```

```
POOL-ASIA-____-____
# put your country licenses for Asia here
SW16080005063000=5wZBpxCwRRucQgRQNipXuSdptwSPgeZxb
twndcFx558Bztq9Ct4rCo/XxqqhDr200FrZYPjeg9T2NTOxYV+
xSCTejMVSS+9WUDFzu+JzGIBEIbZotvNiCL0118bBWr4vAovsp
8ZSi jdCZNXZrCk j2xK5N1VqEDa21L3cJ71vHPHV9817zvnA9ma
Qy8JvXw2fbOchGULnEs6gLwtleoQ4gEmVMeoMmh0z14BqpR/Wc
gi6Wu0T9j9YU8S7WpHZjA1UFapL6etbkCMBWSMnRw==
```

```
POOL-EURO-PE__-____ POOL-ASIA-____-____
# put your global licenses here
SW26080005102000=71MW/SvMfTlNacAkiizdDUz1Fc0BGLKhA
CPmPwwrjT+uS6EVWovzLm7eAtHDhVhfi jSzPj jDzdrO6VH41XZ
1VFvIkRugmsQ0TL7BAJBsmoWRqy9e6s3NMVCRAFbyK00RcXfH1
rbHKQaNzxx3qcn85Pls+LfiHXm5fSPyycbvPx7zgX5h6jE/kS4
nfSLcf5mnL0rDzI3p1LNnsUFJ980e8h/ZmhGxN+9xgVZXiKxdG
XGISMOqt+HSLmc4XQ9RebjqlURE+h1Gu+iqNg4Imw==
```

Note that the same license should occur only once in `licdfile.txt`. If there is a duplicate, a warning is given and the last duplicate in the file is taken.

The local license server gives out seats based on the order of the licenses in `licdfile.txt`, the topmost license first.

Selecting a pool name in `licopt.txt`

To use a pool name, you must make the following modification in file `licopt.txt`: replace the license key by a pool name.

For example in a TriCore product `licopt.txt`:

```
# If you use a local license server you can
# join or group licenses by using a
# pool name instead of a license key.
TSK_LICENSE_KEY_SW160800=POOL-tric-standard
```

For example in a SmartCode product `licopt.txt`:

```
TSK_LICENSE_KEY_SW260800=POOL-EURO-PE__-____
```

TASKING License Management

When the product validates its license the local license server resolves the license request by scanning through all of its licenses in the specified pool.

Note that the 'PRODUCT' keyword, which was present in older versions, is only supported for backwards compatibility.

4.4. Wait for a Free License

In TASKING products built with TLM v1.5r1 or higher (released after March 2023), you can set the `TSK_LICENSE_WAIT` option in `licopt.txt` to a configurable time period to wait for a free license seat to become available instead of directly terminating when a license request is denied.

To wait for a free license, set the license retry time-out (default: none) in `licopt.txt`.

```
TSK_LICENSE_WAIT[=sec]
```

When a request for a floating license is denied because no free seats are available or the license server is unresponsive then the license request(s) will be retried for a maximum of `sec` seconds or until a seat becomes available. If `sec` is 0 or `=sec` is omitted, an unlimited number of retries will be used.

In TASKING products released before March 2023, the TLM licensing system does not include a feature to wait for a free floating license seat. If all available license seats are in use, the licensing system will issue an error message indicating that no free license seats are available, and the build will not start or continue.

To avoid a build failure, you can create a batch file (Windows) or a shell script (Linux) that contacts the TLM license server, for example, every 15 seconds to check for a free license seat until one becomes available. An example content of the batch file for Windows is provided below:

```
:loop
ctc -V
if %errorlevel%==1 goto build
timeout 15
goto loop
:build
echo add makefile call here
```

This batch file example is for a use case where the `PATH` environment variable already includes the path to the directory containing the C compiler executable. The example is for the TriCore product, and the C compiler executable name is `ctc.exe`. If the `PATH` environment variable does not include the entry for the `ctc\bin` sub-directory where `ctc.exe` is located, the path entry must be added to the invocation. For example:

```
"C:\Program Files\TASKING\TriCore v6.3r1\ctc\bin\ctc" -V
```

You can combine the wait for a free license with the license queuing mechanism of the local license server. See [Section 4.5, Using a License Queue](#).

4.5. Using a License Queue

Starting with v2.2r1, the local license server supports a license queue mechanism. When all seats are in use, a license request can be put in a queue. This works on a First In First Out (FIFO) basis. When a free seat becomes available, the first one waiting in the queue will get the license. You can turn license queuing on globally or per license. By default, queuing is off.

To turn queuing on globally

- In `licdopt.txt` add the define:

```
LICD_QUEUE_MODE=ON
```

To turn queuing off globally

- In `licdopt.txt` add the define:

```
LICD_QUEUE_MODE=OFF
```

or remove the define, since OFF is the default behavior.

To overrule global queuing

- In `licdfile.txt` add the define:

```
QUEUE=ON
```

or

```
QUEUE=OFF
```

before a license or pool. This turns the queuing on or off respectively for all licenses following the `QUEUE` define, until overruled by another `QUEUE` define.

For example in `licdfile.txt`:

```
QUEUE=ON
# all license below this line use the license queue mechanism
POOL-EURO-PE__-____
# put your country licenses for Europe here
SW16080005063000=4879pXXTSdt6WNMg5YDnkJvxLxFIP+Cqz
/aZB84oPIyFgJrIXymgVd1qM8odSKkFSL04Vdu7IissrsLVJGQ
yw6qDAPK3beos7Kc62UCICJ1SL3oCIFw7QT7CvD7tSNnV6Wsq0
9e380BHqOhtzVNxAjku8R+X7qLwqhttdZ2HhpRnOtN76WY1BT2
4BcDSQoen5N+A5QmY6BrqCrbKwNJpseOGNR5MiXLe+Ijk635md
V/9L0j/vufm0Uvc9XzQfPaTpxqoOR80Vhbo06UHKQ==
```

When no seats are left the first client in the queue will get the seat when multiple clients try to get the seat by using `TSK_LICENSE_WAIT` functionality or by other ways of repeating requests.

4.6. Bulk Installation

If you have to install a product on a large number of computers you may want to distribute one and the same instance of a TASKING product including the license files (`licopt.txt`) to all computers involved without having to execute the install program on each computer. This functionality is called "bulk installation".

To facilitate a bulk installation

1. Install a TASKING product in the regular way as described in [Chapter 2, Licensing TASKING Products](#).
2. Define a pool name in `licdfile.txt` on the computer where the local license server is installed. For example `POOL-tric-standard`, as explained in [Section 4.3, Joining or Grouping Licenses by Using a License Pool](#).
3. Enter the pool name instead of the license key in file `licopt.txt` on the computer where the TASKING product is installed, as explained in [Section 4.3, Joining or Grouping Licenses by Using a License Pool](#). Note that when you just have a single floating license with a number of seats, you can also use the license key itself instead of a pool name.
4. Copy the installed product tree, including file `licopt.txt`, to all computers.
5. On all computers where you copied `licopt.txt` to in step 4, set the environment variable `TSK_OPTIONS_FILE_<Product-Code><version>` to the location of `licopt.txt`. For example:

```
TSK_OPTIONS_FILE_SW160800v6_3r1=install-dir\licopt.txt
```

When the product validates its license the local license server resolves the license request by scanning through all of its licenses in the specified pool.

Note that the 'PRODUCT' keyword, which was present in older versions, is only supported for backwards compatibility.

4.7. Restricting Access to Licenses

With an access file (default: `licdaccess.txt`) you can bind, deny and permit users and systems to licenses. Without an access file no restrictions apply, every user, system or IP address is allowed access. See option `-A` in [Section 3.4, Local License Server Options and Defines](#).

By default, license requests are anonymous. To see which user name and system name are sent with a license request, you need to add the option `TSK_NO_ANONYMOUS` to your local `licopt.txt` file. See [Section 2.5.1, Licensing Options File](#). On Windows also make sure the Windows environment variable `USERNAME` is set to the user name.

After a change in the access file the license server must be restarted.

The following access rules are possible in an access file. See also the [Order of processing](#) below.

bind

```
bind user { all | name } to key | poolname
bind system { all | name } to key | poolname
bind ip { all | ipaddr[-ipaddr] } to key | poolname
```

With this rule the local license server binds one specific license or license pool to a user, a system, an IP address or a range of IP addresses. A subsequent `bind` rule can overrule a previous `bind` rule. Note that the `bind` rules are processed first, before any `deny/permit` rules are applied.

deny

```
deny user { all | name } [ to { all | key | poolname } ]
deny system { all | name } [ to { all | key | poolname } ]
deny ip { all | ipaddr[-ipaddr] } [ to { all | key | poolname } ]
```

With this rule the local license server denies access to all licenses, a specific license or a specific license pool for a user, a system, an IP address or a range of IP addresses. When you omit the `to` part, this is the same as `to all`, meaning that access to any license serviced by the local license server is denied. A `deny` rule can overrule a previous `permit` rule of the same category (user, system, ip).

permit

```
permit user { all | name } [ to { all | key | poolname } ]
permit system { all | name } [ to { all | key | poolname } ]
permit ip { all | ipaddr[-ipaddr] } [ to { all | key | poolname } ]
```

With this rule the local license server permits access to all licenses, a specific license or a specific license pool for a user, a system, an IP address or a range of IP addresses. When you omit the `to` part, this is the same as `to all`, meaning that access to any license serviced by the local license server is permitted. A `permit` rule can overrule a previous `deny` rule of the same category (user, system, ip).

borrow deny

```
borrow deny user { all | name } [from { all | key | poolname }]
borrow deny system { all | name } [from { all | key | poolname }]
borrow deny ip { all | ipaddr[-ipaddr] } [from { all | key | poolname }]
```

With this rule the local license server denies a user, a system, an IP address or a range of IP addresses to borrow any license or a specific license or a license from a license pool. When you omit the `from` part, this is the same as `from all`, meaning any license. A `borrow deny` rule can overrule a previous `borrow permit` rule of the same category (user, system, ip).

borrow permit

```
borrow permit user { all | name } [from { all | key | poolname }] days n
borrow permit system { all | name } [from { all | key | poolname }] days n
```

TASKING License Management

```
borrow permit ip      { all | ipaddr[-ipaddr] } [from { all | key | poolname }]
                      days n
```

With this rule the local license server permits a user, a system, an IP address or a range of IP addresses to borrow any license or a specific license or a license from a license pool for a maximum number of days. When you omit the `from` part, this is the same as `from all`, meaning any license. A `borrow permit` rule can overrule a previous `borrow deny` rule of the same category (user, system, ip).

Note that the `block` rule, which was present in previous versions, is only supported for backwards compatibility. Use the `deny` rule instead.

Order of processing

The access file is parsed for rules in this order:

1. Parse `bind` rules to optionally enforce a key or license pool.
2. Parse access rules for each category `user/system/ip`. The categories are AND-ed to find the resulting `permit/deny`.
3. Parse `borrow` request rules for each category `user/system/ip`. The categories are AND-ed to find the resulting `borrow permit/deny`.

With the `days` limit for a `borrow`, the last match of each category wins, and the lowest of all categories wins.

Comments

You can add comments to the access file for your convenience, either by a separate comment line or by adding comments at the end of a license access rule. Comments are preceded by the `#` character:

```
# this is a comment
bind user all to POOL-xxxx-yyyy-zzzz # this is a comment behind a rule
```

Examples

To specify that all users are bound to license pool `POOL-xxxx-yyyy-zzzz`, except for users `hans` and `martin` and that they are bound to license key `1234-5678-9012-3456` and `aaaa-bbbb-cccc-dddd` respectively, enter:

```
bind user all to POOL-xxxx-yyyy-zzzz
bind user hans to 1234-5678-9012-3456
bind user martin to aaaa-bbbb-cccc-dddd
```

To bind all users to the (product code) license `3de8-76aa-ed65-3271`, except for two development departments:

```
bind ip all to 3de8-76aa-ed65-3271
bind ip 192.168.1.1-192.168.1.100 to 6a3c-cc42-1b40-a4ec
bind ip 192.168.1.101-192.168.1.200 to 11fa-5eca-7fee-1cd8
```

To allow all access to user `eric` only, enter:

```
deny user all
permit user eric # only allow access to user eric
```

To specify that system `svr100a` is not allowed to access license key `aaaa-bbbb-cccc-dddd`, enter:

```
deny system svr100a to aaaa-bbbb-cccc-dddd
```

To prevent all users except `testuser1` and `testuser2` to access license pool `POOL-work-from-home`, enter:

```
deny user all to POOL-work-from-home
permit user testuser1 to POOL-work-from-home
permit user testuser2 to POOL-work-from-home
```

To setup access restrictions for country licenses and global licenses, such as the Europe and Asia example in [Section 4.3, *Joining or Grouping Licenses by Using a License Pool*](#), enter:

```
deny ip all to all
permit ip ip-range-europe to POOL-EURO-PE__-____
permit ip ip-range-asia to POOL-ASIA-____-____
```

where `ip-range-europe` and `ip-range-asia` should be replaced with the IP range in the format *first-last*.

To allow all IP addresses in a particular range except for a small range to access license key `1111-2222-3333-4444`, enter:

```
permit ip 192.168.1.1-192.168.1.200 to 1111-2222-3333-4444
deny ip 192.168.1.100-192.168.1.120 to 1111-2222-3333-4444
```

To allow an IP address range to borrow a license from `6a3c-cc42-1b40-a4ec` for a maximum of three days, enter:

```
borrow deny ip all from all
borrow permit ip 192.168.1.1-192.168.1.100 from 6a3c-cc42-1b40-a4ec days 3
```

