

License Management Guide

Abstract

In this document you will learn how to configure the Floating License Server (FLS) using the License Manager (Microsoft Windows®) respectively `flsd.conf` (Linux®), how to check out a permanent license key file (Windows only), as well as how to utilize `flsutil` for troubleshooting.

Please note: The license mechanism needs one TCP port for the communication between the client application like TESSY and the Razorcat Floating License Server (FLS). So, please setup your security solutions accordingly. Please read this application note to find out how to set the port on the client and on the server side.

Please note: To setup the Floating License Server (FLS) on a Microsoft Windows® server machine, which is utilized by remote TESSY clients, it is required to start the license manager with sufficient **Administrator privileges!**

Table of Contents

Abstract	1
1 Introduction.....	3
2 Using a Remote Floating License Server	3
3 Setup a Local License Server.....	3
3.1 Request an evaluation key	4
3.2 Register the license key file.....	5
4 Checkout a Permanent License Key File (Windows only)	6
5 Special Features of the Floating License Server (FLS)	6
5.1 Disable DNS lookups	6
5.2 Set a time limit for permanent keys (Windows only).....	6
6 License Manager Expert Tools.....	7
7 Using flsutil	7
8 Access Control Lists (ACLs).....	9

8.1	Introduction	9
8.1.1	Who needs ACLs and why?	9
8.1.2	ACL Scope	9
8.1.3	LDAP – flsldap	9
8.1.4	Using LDAPS (SSL for LDAP).....	10
8.1.5	Configuration Files	10
8.2	Configuration	10
8.2.1	ACL Configuration File	10
8.2.2	ACL File	11
8.2.3	LDAP Configuration Settings.....	11
8.3	Limitations	12
8.3.1	No automatic kick-out of connected clients	12
8.3.2	Permanently checked out licenses	12
9	Continuous Integration License (CI License).....	12
9.1	Configuration	12
9.2	Acquire a CI license	13
10	Migrate from an Older FLS	13
11	Troubleshooting	15
11.1	Failed to start the server.....	15
11.2	Failed to connect to the server	16
11.3	Checking out of permanent keys is always disabled	16
11.4	The License Manager opens although already configured.....	17
11.5	The License Manager fails to start	18
11.6	License for product 'TESSY' is invalid.....	18
11.7	Timeout during connect to remote Floating License Server	18
11.8	ACL List remains empty when utilizing an LDAP configuration	18

1 Introduction

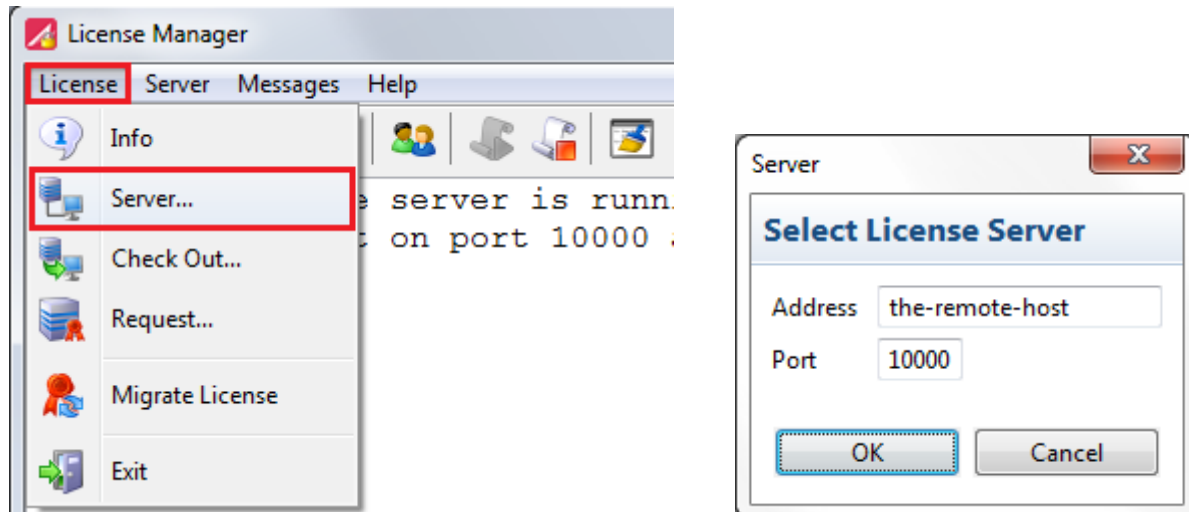
There are two ways to install TESSY's license management tools. You can either download and install the TESSY full setup installer or download and install the standalone FLS full setup. Both installers can be downloaded from

<https://www.razorcat.com/downloads-tesy.html>.

If the installation was successful, it is time to setup the Floating License Server (FLS). Depending on where it runs—remote or local—you may skip chapter 2 or 3.

2 Using a Remote Floating License Server

On **Microsoft Windows®**, the first time TESSY is started, the License Manager will be launched. It displays your local host Id in conjunction with a language Id and a distributor code. Since you do not need a new license file for your local license server click **Done** and select **Cancel** from the next dialog. All you have to configure is the remote license server's name and port number. Select **License** from the License Manager menu and click **Server....**



Enter the remote host's name into the **Address** field and the remote license server's port number into the **Port** field. Ask your administrator for the correct values. Finally click **OK**. Unless errors occur, close the License Manager and launch TESSY now.

On **Linux®**, the variable **Host** found in file `$HOME/.razorcat/.fls/fls.conf` points to the floating license server's name and the variable **Port** holds the destination's port number to be used. There is also a system wide file where these variables can be set, which is found in `/etc/opt/razorcat/fls/fls.conf`.

Please note that TESSY always prefers a locally running FLS. So if you want to use a remote FLS, you have to stop any locally running FLS.

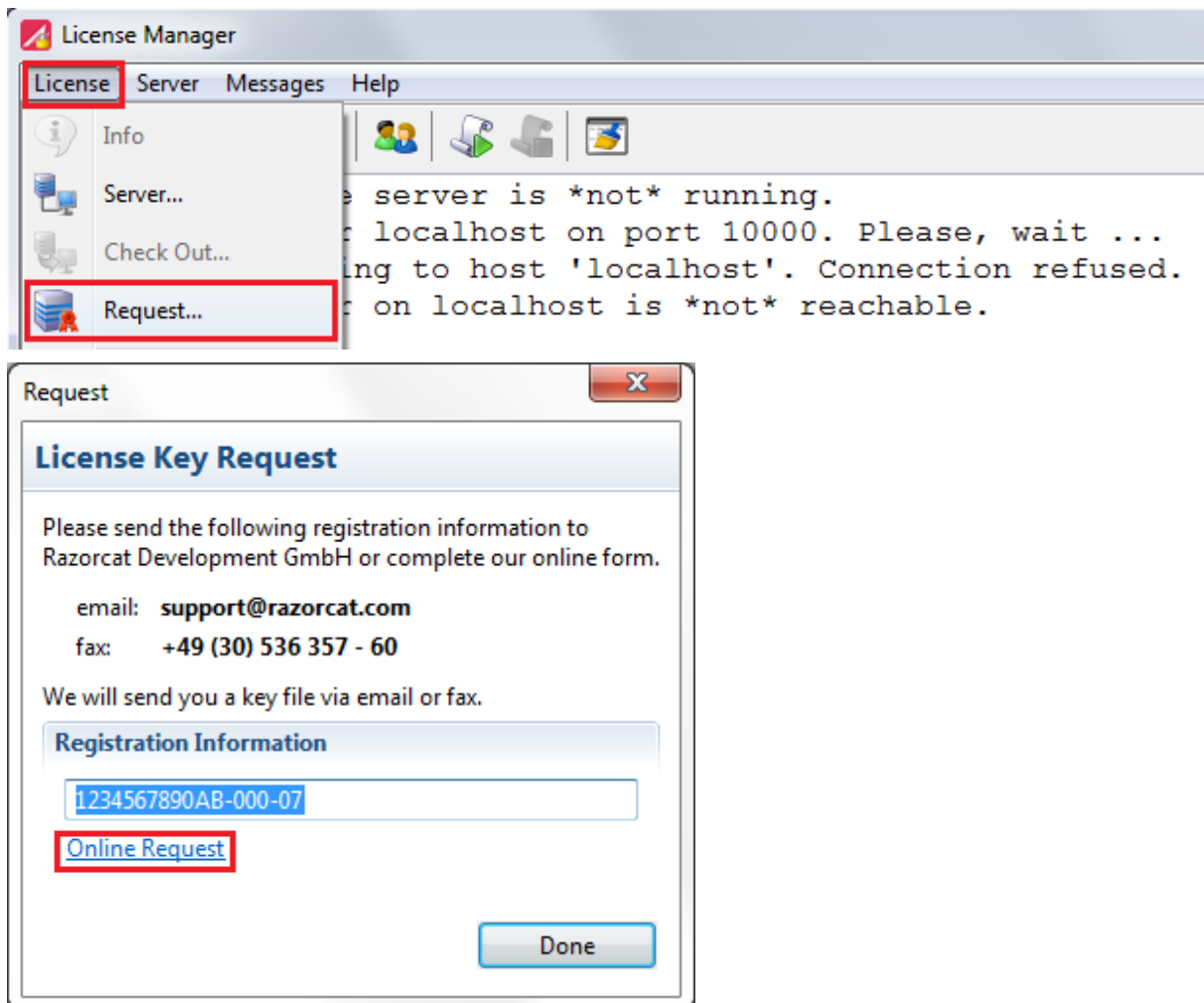
3 Setup a Local License Server

On **Microsoft Windows®**, please launch the Floating License Manager from your Windows menu. You do need **Administrator privileges** if the server shall be a floating license server.

On **Linux**®, the FLS reads its system environment variables from file `flsd.conf` which is found in `/etc/opt/razorcat/fls/flsd.conf`. It expects one entry per line. Empty lines are ignored. A comment starts with an `#` and ends at the end of the same line. Use `flsd --help` to display all supported variables.

3.1 Request an evaluation key

On **Microsoft Windows**®, if you already received a license key file, you can skip this subchapter. Otherwise, click **License** from the License Manager menu and select **Request...** in order to find out your host Id respectively automatically open the online request page.

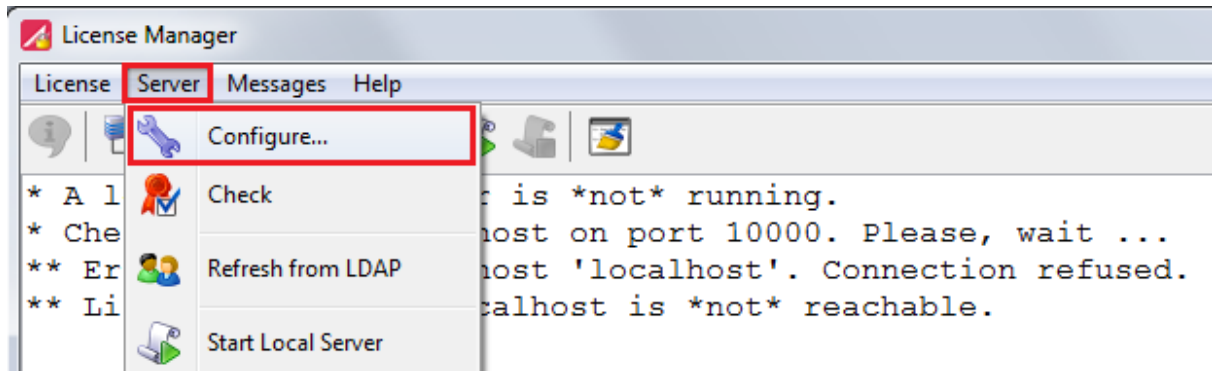


Click **Online Request** which will launch your default web browser opening the *evaluation key request* page. Fill in the form and click **Submit** to request the evaluation license key file. Only company and university e-mail addresses are accepted. You should receive the evaluation license key file during the next working day.

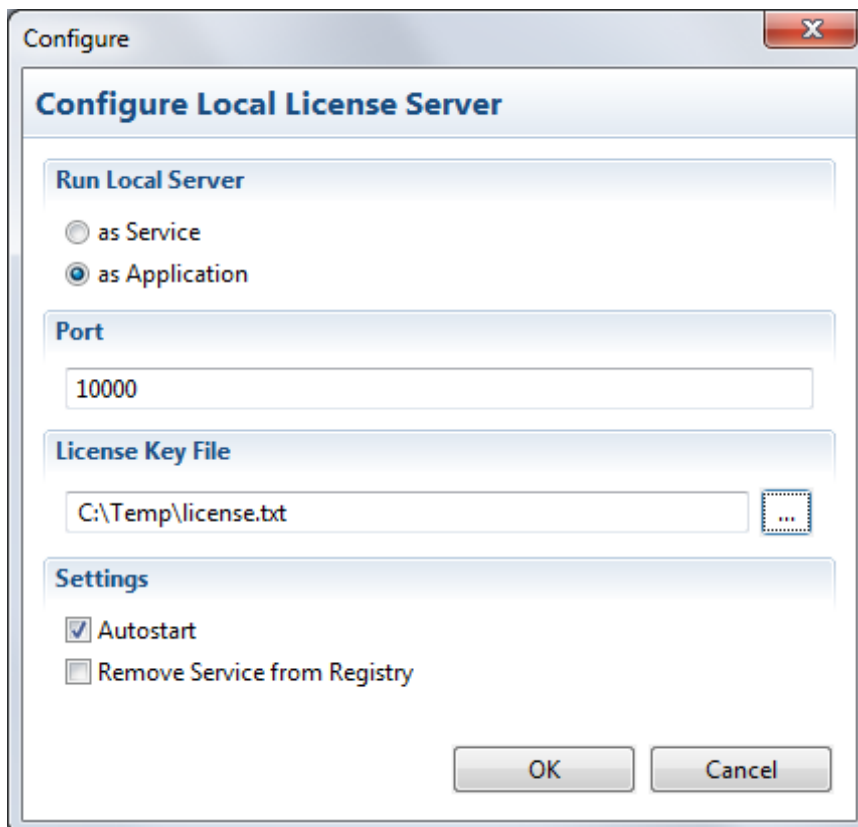
On **Linux**®, execute `/opt/razorcat/fls/bin/flsutil request-license` from command line, which opens the corresponding URL having the host Id already correctly filled in.

3.2 Register the license key file

From the License Manager menu select **Server** and click **Configure...**



Enter the absolute path to your license key file by clicking ... and browsing to the license key file using the **Open** dialog.



Please decide for yourself if you would like to run the FLS as a service or as an application. It is recommended to select **Autostart**. Otherwise TESSY will always launch the License Manager if no remote FLS was found and also no local FLS is running. Finally, click **OK**. If **Autostart** was selected, the License Manager will try to launch the FLS. Unless errors occur, you may close the License Manager now.

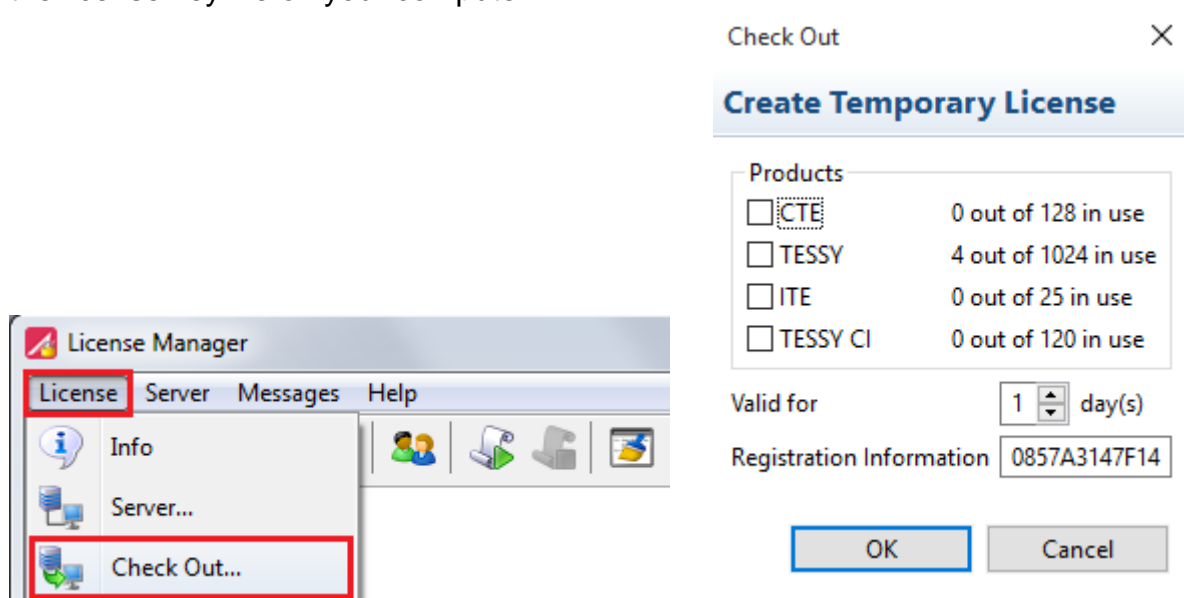
On **Linux**[®], edit the file `/etc/opt/razorcat/fls/flsd.conf` and fill in the variables appropriately.

4 Checkout a Permanent License Key File (Windows only)

Start the License Manager, select **License** from the menu, and click **Info** to find out if the FLS allows checking out of permanent licenses and, if so, how long, e.g.

State.....: Checking out of license keys is enabled--up to 3 days

Select **License** from the License Manager menu and click **Check Out...** Select the number of days, enter *the host Id the license shall be used for*, click **OK**, and save the license key file on your computer.



Finally, register the saved license key file as described in subchapter 3.2 using the FLS on your local computer. **Please note that a permanent license key cannot be given back!**

5 Special Features of the Floating License Server (FLS)

You should not need to start the FLS from command line unless you have to trace and fix some errors occurring during the setup of your FLS configuration. Normally it should suffice to configure and start the FLS as described in chapter 2 respectively in chapter 3. However, the FLS provides a few more features which might be helpful to cope with some problems.

5.1 Disable DNS lookups

By default, the FLS tries to resolve the client's host name. Thus, when utilizing **Info** from the License Manager, all connected clients are listed by their user and host names. Since version 8.2.4 the name resolution is done in parallel. So, disabling this feature is needed only, if you do not want to display the host names instead of the IP numbers. Set the environment variable `FLSD_90_FLAGS` to the value of `--no-dns` if you want to display IP numbers only.

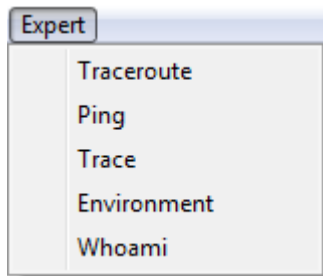
5.2 Set a time limit for permanent keys (Windows only)

By default, the FLS allows checking out of licenses for up to 30 days, which is also the maximum limit. It is possible to decrease this limit or even disable it at all for remote hosts. Set the environment variable `FLSD_90_KEY_DAYS` to any number

from 1 up to 30 to allow remote hosts to checkout a permanent license key for the given number of days. Set the environment variable `FLSD_90_FLAGS` to the value of `--no-keys` in order to disallow checking out of permanent license keys at all. The time limit or disabling of permanent keys holds for remote hosts only, i.e. you may still checkout a permanent license key if the License Manager was started on the same server from which the FLS was started.

6 License Manager Expert Tools

The Floating License Manager (FLS) supports several tools, which may help to solve problems concerning the communication between the Floating License Server (FLS) and its clients like TESSY. These tools comprise running a traceroute to the configured FLS, sending a ping to the configured FLS, tracing live log messages from the locally running FLS, listing the local environment variables, and executing a `whoami` for the locally running FLS. To utilize these tools, please press `Control+Shift+E` to extend the menu of the License Manager by a new menu which is called **Expert**.



7 Using flsutil

`flsutil` provides some convenient features which allow you to check the configuration, test the connection to your Floating License Server (FLS), shut down the local FLS, start and stop statistics logging from command line, display *Info* on command line, fake a license checkout (Windows only), receiving log messages from the running FLS using the `trace` option, and create an FLS support file. The latter is different from a TESSY support file. The following commands are supported by `flsutil` from command line.

help

Display the list of all supported commands next to a brief description of each command.

check-license (Linux only)

Check the integrity of the currently installed license.

checkout [TESSY|CTE|ITE|CCDL|TESSY_CI] (Windows only)

Try to check out the given license type (default is TESSY). You may also select a different host using option `-h` and/or a different port using option `-p`.

config

List the absolute path to the FLS configuration file, the default host name to be used for the FLS, and its default remote port number.

info

Display same output as **Info** does within the License Manager.

install-license license_file (Linux only)

Install the given license file into the location pointed to by **FLSD_90_FILE**. This command requires root privileges.

logfile path_to_log_file [-h localhost]

Enable verbose logging to the given logfile. This feature is only available for the locally running FLS. Use `flsutil info` to check the current logging status.

logstat [-h localhost]

Enable usage statistics logging. The license server creates a usage statistics log file in the `bin` folder of the FLS installation directory named `license-usage.log`. If the configuration contains a fallback FLS, you will have to add `-h localhost`. This command is supported for the local host only.

nologfile [-h localhost]

Disable verbose logging to a file. If it was disabled before, nothing changes.

nologstat [-h localhost]

Disable usage statistics logging. If the configuration contains a fallback FLS, you will have to add `-h localhost`. This command is supported for the local host only.

request-license (Linux only)

Open the license request page using the default browser or, if this fails, display the corresponding URL.

shutdown [-h localhost]

Shut down the locally running FLS. If the configuration contains a fallback FLS, you will have to add `-h localhost`. This command is supported for the local host only.

support [-h|-p]

Create an FLS support file (`flsutil.log`) within the current working folder.

test [-v|-h|-p]

Test the network connection to the given (`-h <host name>`) or default host name with a given (`-p <port number>`) or default port number. It is recommended to turn on verbose mode (`-v`).

trace [-h localhost]

Trace log messages from the locally running FLS to standard output.

update

Triggers the FLS to check for an ACL update trigger file immediately. By default, the FLS checks every 10 Minutes for an ACL update trigger file, if the ACL feature was enabled at startup (see chapter 8.1.2).

8 Access Control Lists (ACLs)

8.1 Introduction

8.1.1 Who needs ACLs and why?

Suppose your company has several software test departments around the globe, each requiring a specific amount of TESSY licenses while each department has ordered its own bulk of TESSY licenses and all licenses should be handled by a single floating license server. Let us say, department A has ten employees around the globe, so that they ordered just six floating licenses, because they suppose that they do not need more than six licenses at the same time. Department B has eight employees all working at the same location and thus ordered eight floating licenses. Furthermore, there are some external employees varying in numbers who need maybe at most five floating licenses. So, in sum you would need $6 + 8 + 5 = 19$ floating licenses. As time goes by, department A expands and uses more licenses as ordered before. As a result, department B complains about the lack of licenses and demands its own bulk of ordered licenses but does not know whom to blame for it. So, to avoid such quarrel the FLS provides some sort of access control lists.

8.1.2 ACL Scope

The FLS access control lists provide an instrument to restrict the access to a license based on the user name (and optional host name) of the computer running the TESSY client and based on a fixed contingent number configured for each ACL. The contingent numbers of the ACLs are summed up. If a contingent number is larger than the amount of the remaining product licenses, then the ACL group is skipped. If there are more license available than the sum of all ACL contingents requires, then the rest is available as normal licenses without any further restrictions.

The ACL files can be created manually by you or automatically by the `flsldap`, which reads the user data from your LDAP server based on the ACL's specific LDAP configuration file. The FLS updates its internal ACLs on the fly if an ACL file content changes and a special flag file exists, which triggers the update. By default, the check for an ACL update trigger file is done every 10 minutes.

8.1.3 LDAP – flsldap

On **Microsoft Windows®**, the `flsldap.exe` has to be started by a cronjob with argument `--cronjob`. The program launches the `flsldap.jar` Java program. By default, the latter reads the user data only once from the LDAP server and exits afterwards. So, you may trigger it manually by utilizing the *Refresh from LDAP* button of the FLS License Manager. It is also possible to let the program run permanently reading user data regularly based on the given cycle time (see 8.2.3). The additional argument `--keep-running` to the `flsldap.exe` is required to enable this feature. A cycle time of 1 means that the program will contact the LDAP server once per hour and check for updates based on the ACL's LDAP configurations. If the new ACL files differ from the current ones, the program sets a write flag for each differing file and substitutes the current ACL files for the new ones. It finally removes the write flags and creates the specific update flags, so that the FLS will be triggered to update its corresponding internal ACLs, which it does at startup and thereafter

every 10 minutes (by default) if an ACL update trigger file was found. When using argument `--keep-running` the cronjob should be configured as 'at system startup'.

On **Linux**[®], you can utilize the shell script file `flsldap.sh` to run the `flsldap` once and exit afterwards. The shell script uses the ACL configuration file found in variable `FLSD_90_ACL_CONF` of file `/etc/opt/razorcat/fls/flsd.conf`.

Please note: The specific settings for an ACL's LDAP configuration have to be provided by the administrator of your LDAP server. Since those settings are usually unique, Razorcat cannot provide these specific settings!

8.1.4 Using LDAPS (SSL for LDAP)

It is also possible to utilize SSL in LDAP, i.e. an LDAPS connection. All you have to do, is to add the corresponding certificate to your Java keystore. The following three paragraphs show how to add, delete, and list the certificates.

Add a certificate to the keystore

```
keytool -keystore ..\lib\security\cacerts -storepass changeit -trustcacerts
-import -noprompt -alias my-root-ca -file "absolute\path\to\certs\my-root-
cert.cer"
```

```
keytool -keystore ..\lib\security\cacerts -storepass changeit -trustcacerts
-import -noprompt -alias my-intermediate-ca -file
"absolute\path\to\certs\my-intermediate-cert.cer"
```

Delete a certificate from the keystore

```
keytool -keystore ..\lib\security\cacerts -storepass changeit -trustcacerts
-delete -alias my-root-ca
```

```
keytool -keystore ..\lib\security\cacerts -storepass changeit -trustcacerts
-delete -alias my-intermediate-ca
```

List all certificates from the keystore

```
keytool -keystore ..\lib\security\cacerts -storepass changeit -list -v -
alias my-root-ca
```

```
keytool -keystore ..\lib\security\cacerts -storepass changeit -list -v -
alias my-intermediate-ca
```

8.1.5 Configuration Files

The ACL mechanism requires at least the ACL configuration file (see 8.2.1) and one ACL file. Furthermore, each ACL can be related to an LDAP configuration file. Please note that the ACL files have to reside within a folder which is writable by the `flsd` (and the `flsldap` respectively `flsmgr` if the LDAP feature is utilized).

8.2 Configuration

8.2.1 ACL Configuration File

This subchapter describes the ACL configuration file. There is no naming convention regarding the name of that file. The `flsd` as well as the `flsldap` read the file name

and its path from the system environment variable `FLSD_90_ACL_CONF`. So, you may name it for instance `acl.conf`. The file has the following specifications.

- A comment starts with an `#` and ends at the end of the same line.
- Empty lines are ignored.
- Each non-empty line consists of an ACL group name, a product name, a contingent number, and the absolute path to the specific ACL file, e.g. regarding our example from the introduction supposing 19 floating licenses for product TESSY:

```
global=TESSY:6:D:\ACLS\global.acl
head_office=TESSY:8:D:\ACLS\headoffice.acl
external=TESSY:5:D:\ACLS\external.acl
```

- Group names are unique and have no further naming conventions, except that they must not contain an `=` in its name. Spaces are ignored.
- There is no naming convention for a specific ACL file name.
- Spaces are generally ignored, except for the path of the ACL file.
- *Important note: ACL files must be unique for each ACL group!*

8.2.2 ACL File

The ACL file contains per line either a user name, e.g. `john.doe`, or a user name and a host name, e.g. `john.doe@company.com`. Spaces and empty lines are completely ignored, letter case is ignored, a comment starts with an `#` and ends at the end of the same line. The file has to reside in a folder which is writable by the `flsd` (and the `flslldap/flsmgr`). *An ACL file refers to exactly one and only one ACL group!*

8.2.3 LDAP Configuration Settings

Whenever you change the ACL configuration file or one or more of the LDAP configuration files, you should at first recreate the ACLs by utilizing the Refresh from LDAP button of the FLS License Manager!

An LDAP configuration file is bound to an ACL group by adding a further line to the ACL configuration file (see 8.2.1), which has the format, i.e. ACL group name, LDAP, and absolute path to the group's specific LDAP configuration file, for example

```
global=TESSY:6:D:\ACLS\global.acl
global=LDAP:D:\ACLS\global.ldap
head_office=TESSY:8:D:\ACLS\headoffice.acl
head_office=LDAP:D:\ACLS\headoffice.ldap
external=TESSY:5:D:\ACLS\external.acl
external=LDAP:D:\ACLS\external.ldap
```

There is no naming convention for the file name of the LDAP configuration file. Empty lines are ignored. A comment starts with an `#` and ends at the end of the same line. The file has to be readable by the `flslldap.jar` respectively `flsmgr.jar` program. The LDAP configuration file has the following specifications. There are seven attributes which can be configured.

- `provider_url` is *required* and points to the provider URL, e.g. `ldap://ldaphost:389/`
- `sec_principal` is optional and holds the security principals required to obtain the user data

- `sec_credentials` is optional and holds the security credentials, i.e. for instance the password
- `base_name` is *required* and holds the named context or object for entries to search for
- `filter` is *required* and holds the filter expression to use for the search
- `attribute_ids` is optional and holds a comma separated list of attributes to be returned from the search. The content of the first attribute is used as the user name (or user name and host name) for the ACL file. By default, the content of `uid` respectively `sAMAccountName` is used as the user name.
- `cycle_time` holds the period in hours between two LDAP request launches. If this value is 0, which is the default value, the LDAP request is done only once at startup. To enable the feature (i.e. `cycle_time > 0`) the `flsldap.exe` has to be launched with option `--keep-running`. The cronjob started as 'by system startup' would use `'--cronjob --keep-running'` as arguments. *The order of the program arguments is important!*

Let us assume you would like to filter all TESSY users from the LDAP server, which has an already configured group `tessy`, and whose user attribute has the format `first_name.family_name`. Then the `filter` and `attribute_ids` may look like

```
filter = (&(uid=*.*) (memberOf=*cn=tessy*))
attribute_ids = uid,memberOf
```

8.3 Limitations

8.3.1 No automatic kick-out of connected clients

Although the ACLs will update on the fly, existing, i.e. connected clients, which are no longer allowed by the ACL, will not be kicked out of the floating license server. New connections from those clients, however, will be blocked. So, to forcefully kick out such clients, you will have to restart the floating license server using for instance the license manager.

8.3.2 Permanently checked out licenses

At the start of the floating license server those licenses are tried to sort into the appropriate ACL groups. However, if no matching group is found, each of these licenses is sorted into the first group, which has unused licenses.

9 Continuous Integration License (CI License)

This type of license is meant for continuous integration using `tessyd`, e.g. by utilizing a Jenkins server. *The TESSY IDE cannot utilize this type of license.*

9.1 Configuration

You will need at least version 8.2.6 of the Razorcat floating license server. From the perspective of the license server, there is no difference to other licenses the server can handle. All you have to do is to acquire the CI licenses and restart the license server using the new license file.

On the other hand, from the client perspective there are several issues to take care of. In order to enable this feature for `tessyd` of TESSY version 4.1 beginning with version 4.1.34, you will have to start the `tessyd` with argument

```
--enable-CI-license
```

to enable the feature and to fall back to a normal TESSY license if a CI license is not available or with argument

```
--use-CI-license-only
```

to enable and force the checkout of a CI license, i.e. `tessyd` will exit, if there is no CI license available. Alternatively, you may edit `tessy.conf` and add the environment variable `TESSYD_CI_LICENSE_FLAGS`, which holds two flags. The first flag enables (1) or disables (0) the feature for `tessyd`. The second flag determines, if `tessyd` shall exit (2), if there is no CI license.

Since TESSY version 4.2.20 respectively 4.3.13 flag 1 is the default setting, i.e. at first `tessyd` tries to checkout a CI license. If there is no such license available, `tessyd` tries to checkout a normal TESSY license.

If you want `tessyd` to checkout a normal TESSY license only, you would set the environment variable `TESSYD_CI_LICENSE_FLAGS` to 0. If you want to enable the CI license feature and force the checkout of a CI license accepting that `tessyd` might exit and not execute the test run, if no CI license is available, then you would set environment variable `TESSYD_CI_LICENSE_FLAGS` to 3.

In summary	
<code>TESSYD_CI_LICENSE_FLAGS = 0</code>	checkout a normal license only
<code>TESSYD_CI_LICENSE_FLAGS = 1</code>	checkout a CI license if available or fall back to normal
<code>TESSYD_CI_LICENSE_FLAGS = 3</code>	checkout a CI license if available or exit

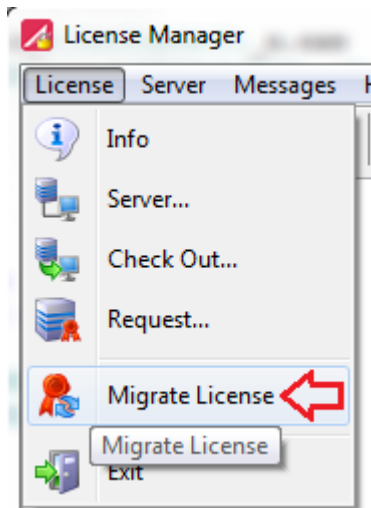
1 is the default value since version 4.2.20 respectively 4.3.13.

9.2 Acquire a CI license

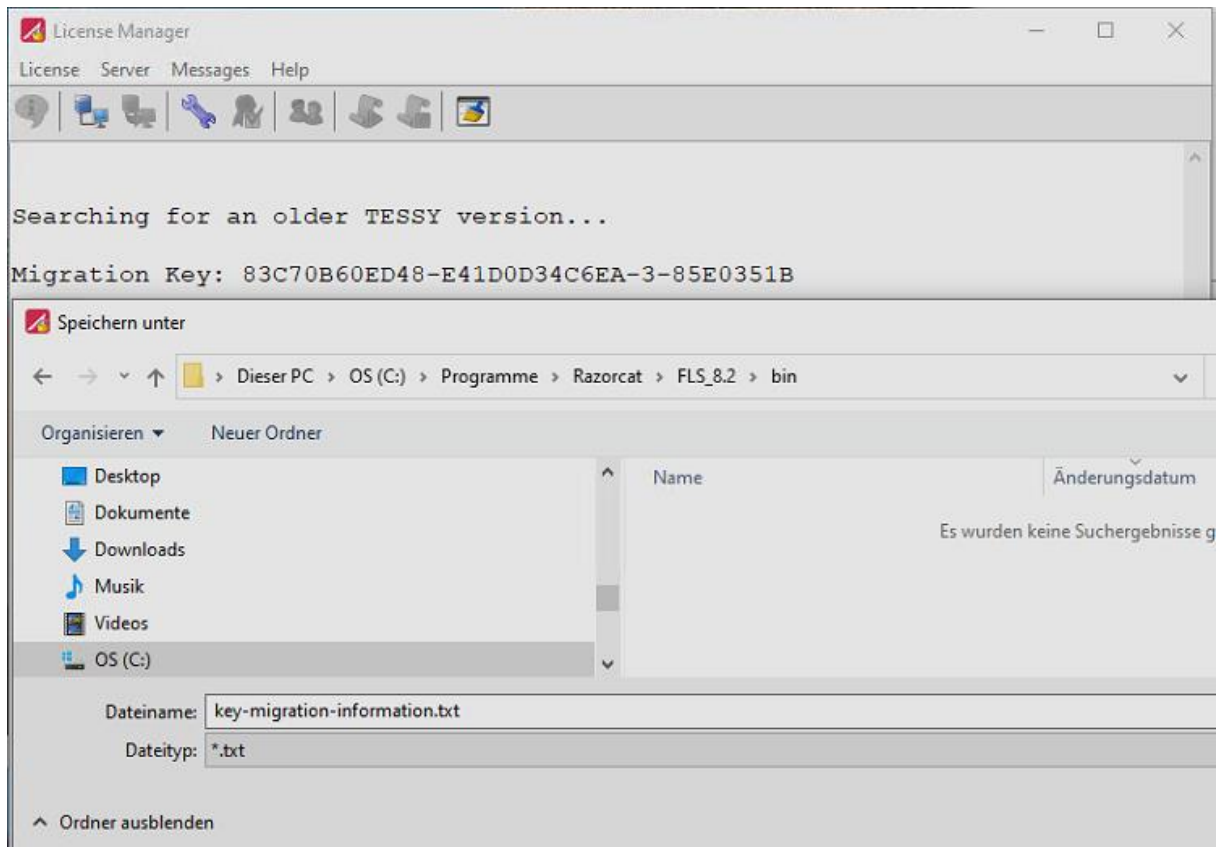
Please contact our sales team at sales@razorcat.com to get more information about the terms and condition and to order a price list.

10 Migrate from an Older FLS

When migrating the Razorcat Floating License Server from a version prior to 8.2.4 to the latest one, you will have to migrate your license key as well. So, please open the Floating License Manager of the newly installed Floating License Server, open the **License** menu and select **Migrate License** as shown below.



The Floating License Manager will create a key migration information file, which you should save into a writable folder.



Send the `key-migration-information.txt` file to migrate@razorcat.com. You will receive a new license key file via E-Mail the next working day. Finally, install the license key file as described in chapter 3.2.

11 Troubleshooting

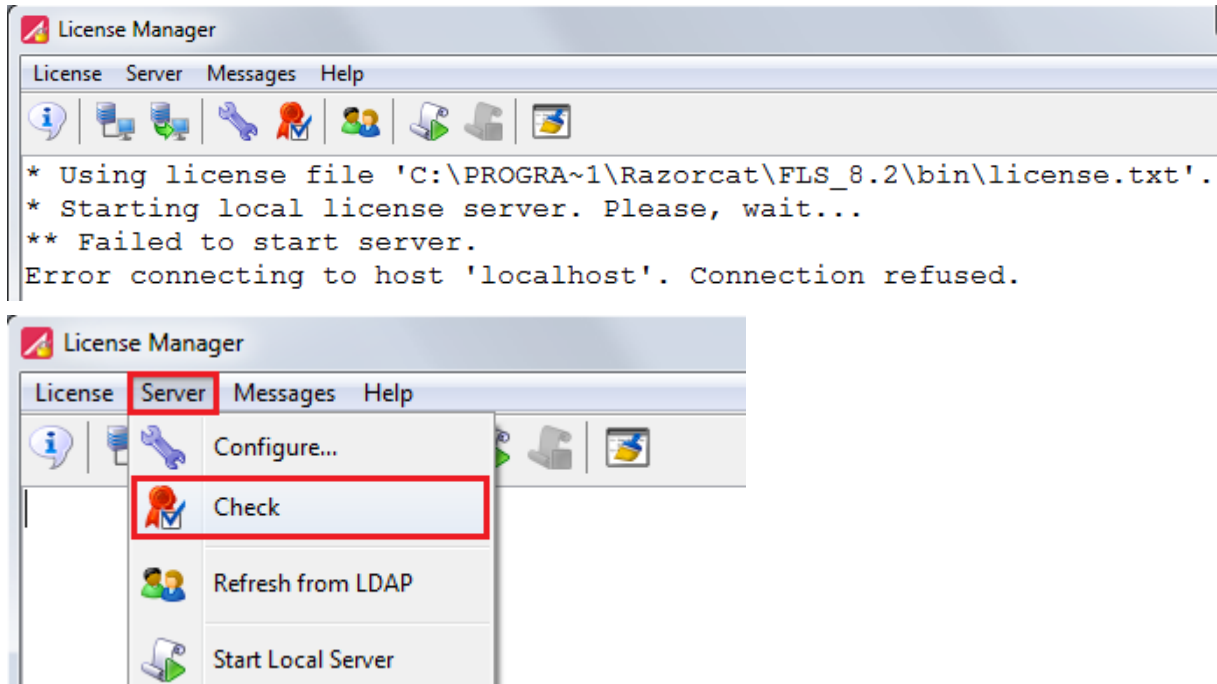
11.1 Failed to start the server

On **Linux**®, open a command shell and enter the following command.

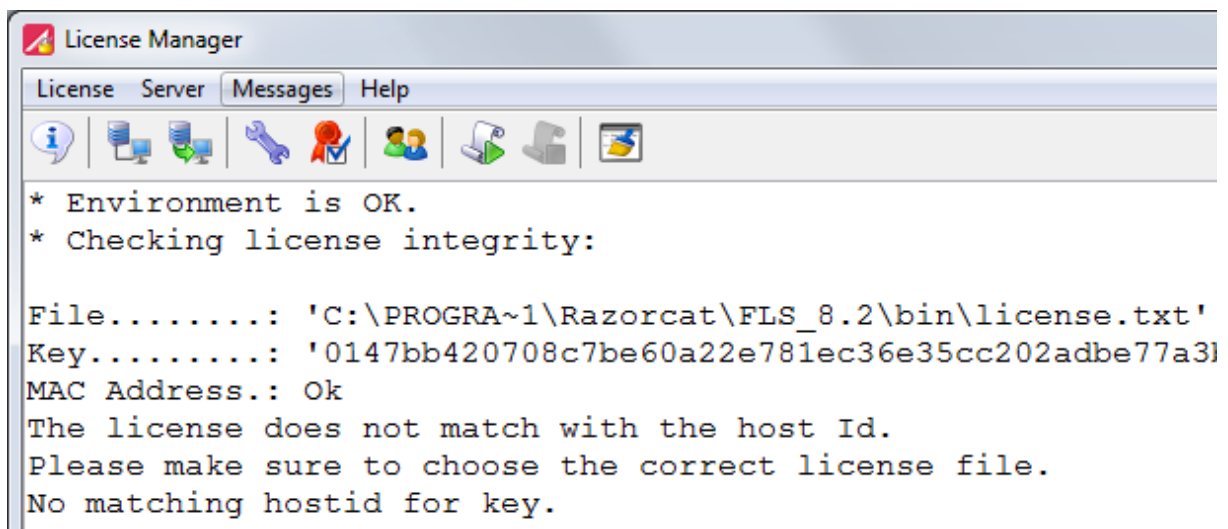
```
journalctl -e
```

Search for `flsd` to find the corresponding error messages.

On **Microsoft Windows**®, please check your license key file by selecting **Server** from the License Manager menu and click **Check**.



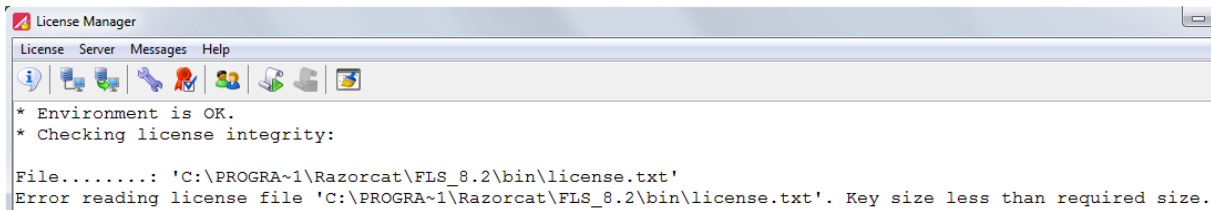
One result might look like the following.



In this case it is most likely that you have chosen the wrong license key file. Please compare the license key file with the one you received via e-mail. If you made sure

the license key file is correct, send the license key file plus your current host Id to support@razorcat.com.

Another result might be the following.



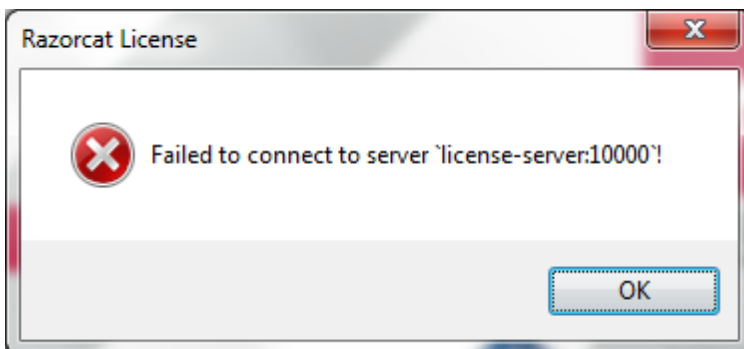
On some server systems copying a file results in an empty file in the destination directory. So please make sure the license key file is the same as the one sent to you via e-mail.

However, if the license key file check succeeds, the content of the license key is displayed. That means that the FLS itself fails to start because it was blocked in any way. So, please open a Windows command shell **with Administrator privileges**, change to the `bin` folder of the FLS installation directory and try to start the FLS from command line.

```
flsd.exe --foreground --verbose --file <path to your license file>
```

Please make sure that no virus scanner or firewall blocks the FLS from startup and that the port it runs on is open to your TESSY clients.

11.2 Failed to connect to the server



If you encounter this dialog after TESSY start up, launch the Floating License Manager and check your address and port number settings as described in chapter 2. *Resolve the machine name where the license server runs on to its IP address using `nslookup` from a Windows command shell. Use the server's IP address as **Address**.* If the settings are correct, select **License** from the License Manager menu and click **Info**. If the license server information is *not* listed or a timeout occurs, check the connection using "`flsutil test -v`" (see chapter 7). Please make sure that no virus scanner or firewall blocks TESSY from contacting the FLS.

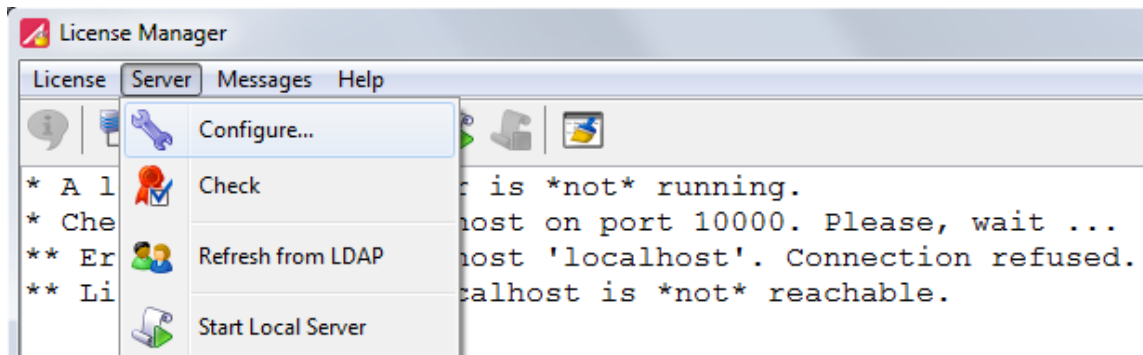
11.3 Checking out of permanent keys is always disabled

The checkout of permanent keys might be disabled deliberately by an environment variable (see subchapter 5.2).

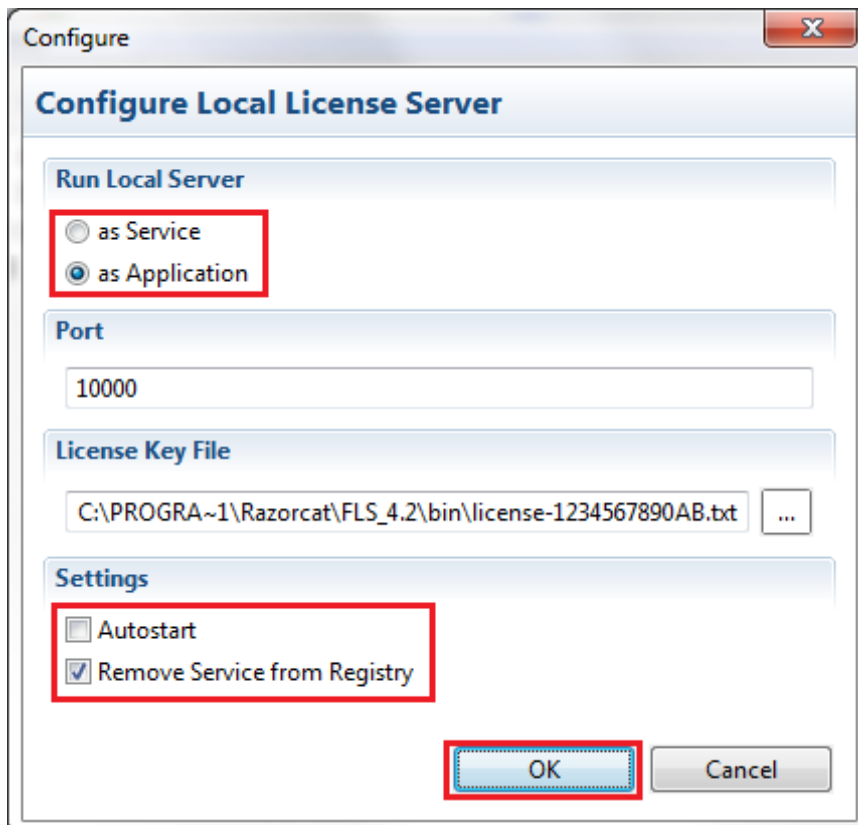
It is also possible that the MAC address of your network interface changed or that your registry was destroyed. If this is the case, you will have to uninstall and reinstall the FLS. That means that the registry will be repaired and a new host Id will be generated. When you start the FLS afterwards, a Repair-Id is displayed, which you may send to repair@razorcat.com in order to get a new license key file. The procedure of sending the new license key file is done manually. Thus it may take a working day.

11.4 The License Manager opens although already configured

You configured a remote FLS but, when TESSY starts up, it still opens the License Manager. The reason might be that you already registered an evaluation license key file on your local computer. So please start your License Manager **with Administrator privileges**, stop the running FLS, select **Server** from the License Manager menu, and click **Configure...**



Now, set the FLS to be run **as Application**, disable **Autostart**, and select **Remove Service from Registry** as shown below. Finally, click **OK**.



11.5 The License Manager fails to start

If **Data Execution Prevention** is enabled on your Windows operating system, the `flsmgr.exe` has to be added to the list of excepted applications. Also a virus scanner or a firewall might be the reason for a failed start up.

11.6 License for product 'TESSY' is invalid

Make sure you have chosen a valid Floating License Server. *Licenses which have been outdated by new licenses become invalid and must not be used any longer!*

11.7 Timeout during connect to remote Floating License Server

The reason might be a misconfigured or slow local DNS server. Please refer to subchapter 5.1 for a solution of this problem.

11.8 ACL List remains empty when utilizing an LDAP configuration

You may set the system environment variable `FLSD_90_LDAP_TRACE` to `1` in order to check what is returned by the LDAP server based on your LDAP configuration file.