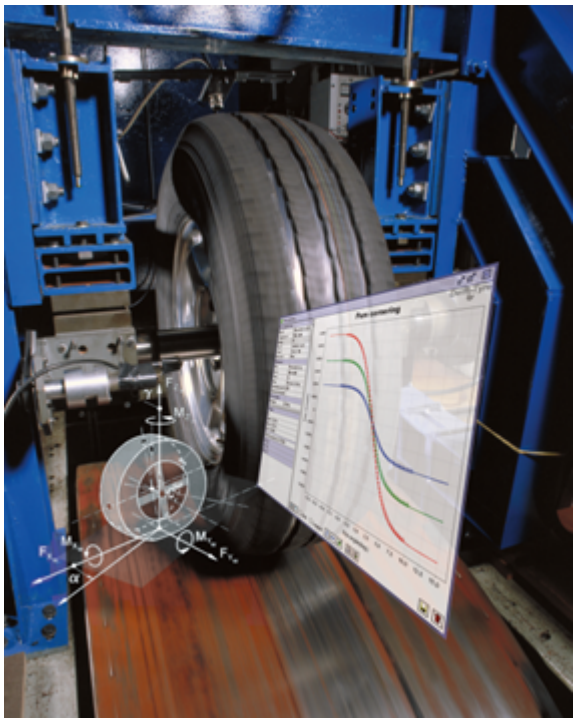
# MF-Tyre/MF-Swift 6.2.0.4

Installation Guide





Document revision: 1-3-2017

Printed: February, 2017, The Netherlands

delfttyre

**Publisher**

*TNO*
*PO Box 756*
*5700 AT Helmond*
*The Netherlands*

delfttyre

# Table of Contents

# Installation Guide

MF-Tyre/MF-Swift is a plug-in to a number of Multi-Body Simulation Packages (MBS) capable of simulating (dynamic) vehicle behaviour. In most cases the MF-Tyre/MF-Swift capability is included in the installation of these MBS packages. This, however, does not hold for the coupling of MF-Tyre/MF-Swift with:

- MATLAB/Simulink (MathWorks)
- ADAMS (MSC)

The coupling with the above mentioned packages is described in this manual.

**Note:** the license manual applies only for MF-Swift users, MF-Tyre is freeware and hence no license server needs to be specified.

As of MF-Tyre/MF-Swift release 6.2 two types of installations are supported, i.e. the regular installation and the Served Model Deployment (SMD) installation. Each having its own installer.

In general the installation of MF-Tyre/MF-Swift consists of the following steps:

- MF-Tyre/MF-Swift installation
- connecting MF-Tyre/MF-Swift to MBS Packages
- licensing MF-Tyre/MF-Swift

The compatibility of MF-Tyre/MF-Swift to various MBS packages is described in the Compatibility Table.

# 1 System Requirements

(Minimum) system requirements for MF-Tyre/MF-Swift are:

- Operating systems: Windows 7 64-bit
- Pentium 4 or higher
- 512 Mb of RAM
- MF-Swift: 100 Mb of disk space

**Note 1**: The MBS Package MF-Tyre/MF-Swift is connected to may have specific system requirements.

# 2 Compatibility Table

The MF-Tyre/MF-Swift models are available for a wide variety of multi-body simulation packages. We may distinguish between:

***Coupling with MBS package done by TNO***

ADAMS (MSC software)
MATLAB/Simulink (MathWorks)

***Coupling with MBS package done by MBS package supplier***

Recurdyn (FunctionBay)
CarSim/TruckSim/BikeSim (Mechanical Simulation Corporation)
Dymola (Modelon) (NOTE: Modelon delivers a general Modellica interface)
AVL software (AVL)
SAMCEF (LMS Samtech)
MotionSolve (Altair)
DAFUL (VirtualMotion)
VI-CarRealTime (VI-Grade)
Virtual.Lab (LMS)
SIMPACK (SIMPACK AG)
MADYMO (TASS)

The corresponding compatibility table is shown below.

| Multi-body package | Version | Win32 ** | Win64 *** |
|---|---|---|---|
| ADAMS*** | 2014 | | x |
| | 2015 | | x |
| | 2015.1 | | x |
| MATLAB/Simulink*** | 2014 and up | | x |
| Recurdyn | * | * | * |
| CarSim/TruckSim/BikeSim | * | * | * |
| Dymola | * | * | * |
| AVL software | * | * | * |
| SAMCEF | * | * | * |
| MotionSolve | * | * | * |
| DAFUL | * | * | * |
| VI-CarRealTime | * | * | * |
| Virtual.Lab | * | * | * |
| SIMPACK | * | * | * |
| MADYMO | * | * | * |

*: Availability and MF-Tyre/MF-Swift version depends on the implementation in the respective multi-body packages.
**: Win32 includes Windows XP, Windows 7 (32-bit and 64-bit)
***: Earlier Matlab and ADAMS versions are supported by MF-Tyre/MF-Swift v6.2.0.3

# 3 Windows Installation

This chapter describes the steps to follow to install MF-Tyre/MF-Swift on a Microsoft Windows system.

## 3.1 MF-Tyre/MF-Swift Installation

**Note**: To install the products you need to have administrator rights on the machine (for installing the program and setting the environment variables).

To install MF-Tyre/MF-Swift run the following application. The setup wizard will guide you through the installation process. On Windows 64-bit editions:

```
setup_MFTyre_MFSwift_6204_x64.exe
```

After installing the tyre models the MF-Swift model needs a license key, the path to the license server will be set during installation. If the license server still has to be configured please follow the steps described in the License Manual 13.
Specifying the license server is only required for MF-Swift users. MF-Tyre is freeware and hence no license server needs to be specified.
The coupling with any supported ADAMS version can be defined during installation. For the coupling with MATLAB please follow the instructions in Connection to MBS Packages - MATLAB 9.

## 3.2 Connection to MBS Packages

As mentioned in 'Installation Guide 5 the coupling to the MBS packages MATLAB/Simulink (MathWorks) 9 and Adams (MSC) 8 is described in this section.

### 3.2.1 Adams

During installation select the requested ADAMS version which you would like to use. The installer automatically establishes the necessary coupling. During installation it is possible to install all supplied interfaces, however only one can be activated.

For each supported version of ADAMS a separate interface is shipped with MF-Tyre/MF-Swift. The directory where these interfaces are located, can be found via the Windows “Start” menu ( TNO DelftTyre > MF-Tyre & MF-Swift 6.2.0.4 > ADAMS > Plugin_library). If you would like to switch to a different ADAMS version please copy the interface from <install_dir>\ADAMS\Plugin_library\<Adams_version> to <installation_dir>\DelftTyre_LIBS manually.

**Note 1:** The interface with ADAMS has changed from "Adams User library method" to "User Tire Library plug-in method" as of MF-Tyre/MF-Swift release 6.2. This means that no TNO specific solvers are used anymore. Note that if you have used earlier releases of the MF-Tyre/MF-Swift models, it is required to remove the acar_solver.dll from the Adams $HOME\acar_private directory; by default this is: %HOMEDRIVE%% HOMEPATH%\acar_private.

**Note 2:** From version 2013 onward ADAMS automatically redistributes the tyre mass. To assure correct behaviour of the MF-Tyre/MF-Swift model set the environment variable MSC_ADAMS_TIRE_DIS_M_AND_I to 'NO', such that ADAMS does not apply a mass re-distribution.

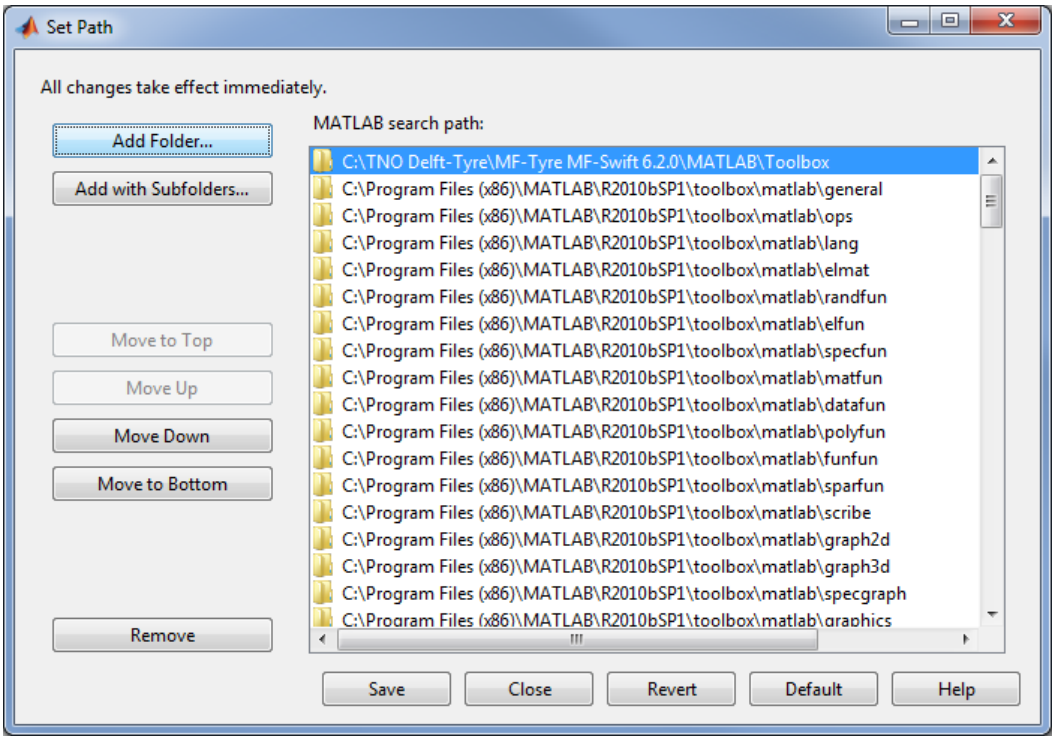
**Note 3:** For ADAMS 2013 onward it is necessary to add a directory to the system search path to be able to run the program from the command line (for details: MSC tech article ID: KB8020901)

Details about how to use the MF-Tyre/MF-Swift model with ADAMS can be found in the MF-Tyre/MF-Swift UserManual.

### 3.2.2 MATLAB

To use the MF-Tyre/MF-Swift model with MATLAB, Simulink and SimMechanics, the "Toolbox" directory has to be on the MATLAB search path. Typically this directory has the name:

```
C:\TNO Delft-Tyre\MF-Tyre MF-Swift 6.2.0.4\MATLAB\Toolbox
```




The following steps should be taken:

- From the MATLAB menu select:
  File > Set Path…
- A new window showing the MATLAB search path will appear (see figure above). In this window click on "Add Folder…"
- Browse for the Toolbox directory, followed by "ok" in the "Browse For Folder" window.

- Finally in the "Set Path" window click on "Save" and "Close". From now on the MF-Tyre/MF-Swift model is available in every MATLAB session.

Some examples of using the tools are given in the Start Menu: All Programs > TNO Delft-Tyre > MF-Tyre & MF-Swift 6.2.0.4 > MATLAB > Simulink demo's (and SimMechanics demo's, respectively).

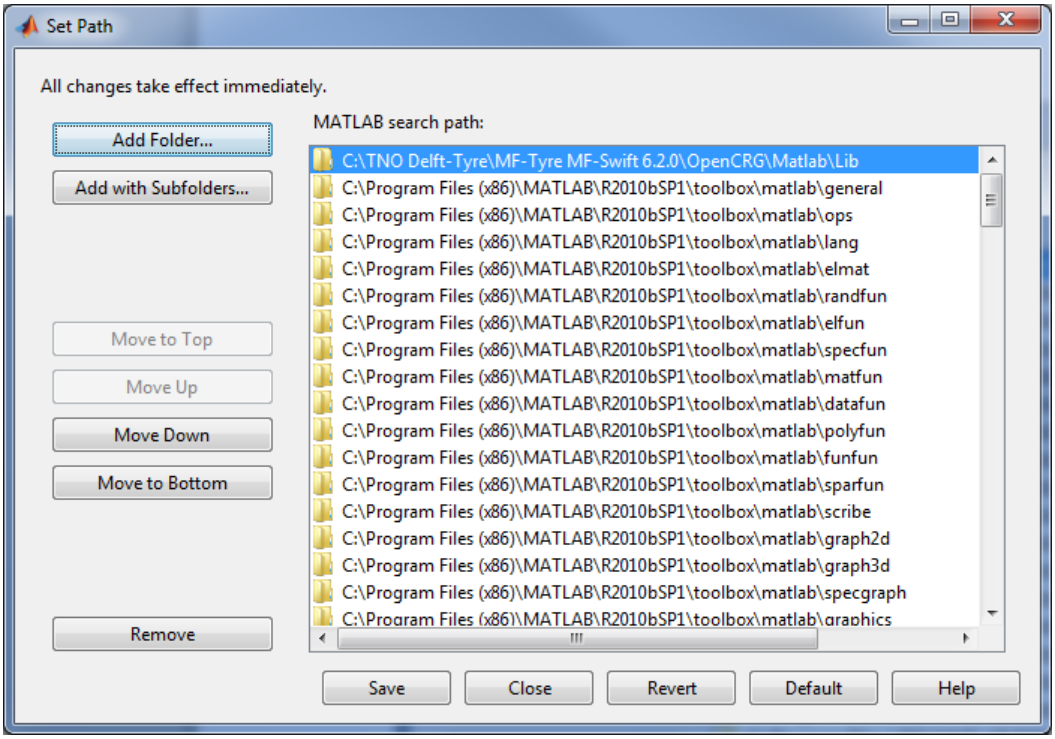
**Note**: Due to the changes of the library layout, the MF-Tyre/MF-Swift interface in already existing Simulink and SimMechanics first generation models need to be updated after the installation of MF-Tyre/MF-Swift 6.2. For details please see the Help section Matlab.

**Note**: If you want to use the OpenCRG routines delivered with the Delft-Tyre products, also add the OpenCRG library 10 to your MATLAB search path.

## 3.3 OpenCRG

If you want to use the OpenCRG routines delivered with the DelftTyre products add the library to your MATLAB search path
Typically this directory has the name:

```
C:\TNO Delft-Tyre\MF-Tyre MF-Swift 6.2.0.4\OpenCRG\Matlab\Lib
```




The following steps should be taken:

- From the MATLAB menu select:
  File > Set Path…
- A new window showing the MATLAB search path will appear (see figure above). In this window click on "Add Folder…"
- Browse for the Library directory, followed by "ok" in the "Browse For Folder" window.
- Finally in the "Set Path" window click on "Save" and "Close". From now on the OpenCRG routines are available in every MATLAB session.

Some examples of using the tools are given in the Start Menu:

*All Programs > TNO Delft-Tyre > MF-Tyre & MF-Swift 6.2.0.4 > OpenCRG > Matlab demo's*.

# 4 License Manual

This chapter contains information regarding the license system necessary to run the Delft-Tyre products. Licensing is used in two ways in this chapter: licensing regarding terms and conditions, and licensing as a mechanism to protect the software from unauthorized use. The context will reveal the meaning.

The terms and conditions governing the licensing of **MF-Tyre** consist solely of those set forth in the document titled 'License conditions of MF-Tyre software'. The terms and conditions governing the licensing of **MF-Swift**, **MF-Tool** and other Delft-Tyre software consist solely of those set forth in the **written contracts** between TNO and its customers.

MF-Tyre/MF-Swift includes OpenCRG, licensed under the Apache License, Version 2.0.

The software is protected / licensed with Flexera. Licensed products include:

- MF-Swift
- MF-Tool

**Note**: For MF-Tyre no license is required.

Note that the license information described holds for Delft-Tyre Release 6.2 products. The licensing tools are distributed via the installer.

In the remainder of this document the following convention is used:
- `installationdir`: The full path of the directory where the Delft-Tyre product is installed, including the version, for example: C:\TNO Delft-Tyre\MF-Tyre MF-Swift 6.2.0.4.

## 4.1 Licensing Set Up

### Available License type

Delft-Tyre is licensed by the feature licenses which allows a specific Delft-Tyre module to run. A single license for the given feature name is checked out when feature based licenses are used. The licensing mechanism supports floating and node-locked licenses. No uncounted node-locked licenses are offered for native Delft-Tyre licensing.

### Obtaining a license

The various licenses for Delft-Tyre products can be retrieved via your Delft-Tyre sales representative. The Delft-Tyre license will be locked to a specific computer (a stand-alone machine or license server). Therefore some information is necessary to identify this computer, which is the host name and MAC address.
On Windows this can be obtained by typing: "ipconfig /all" in a command window.
On Linux the host name can be obtained by typing: "cat etc/hostname" in a terminal and for the MAC address type: "ifconfig".

### Install clients

The Delft-Tyre license server(s) has(have) to be specified by the **MADLIC_LICENSE_FILE** environment variable. This environment variable is used by the applications to find the license server. When no license server is specified via the environment variable, the license server defaults to port 26000 on the local machine (26000@localhost). This is valid for both multiple stack and single stack licenses.
The environment variable can be modified by opening the “Environment Variables” menu of your Windows installation (right click “My Computer”, select “Properties” (and "Advanced System Settings" if using Windows 7), and select the “Advanced” tab. Press the “Environment Variables” button and edit the **MADLIC_LICENSE_FILE** environment variable.

The environment variable **MADLIC_QUEUE_MAX_MINUTES** determines the time an application stays queued if no licenses are available in batch mode. The variable has the following meaning:

- If set to 0, no queuing takes place, the application will terminate
- If not set, the application is queued for 60 minutes where each minute a check is done if licenses have become available.
- If set to any x >0, the application will queue for x minutes where each minute a check is done if licenses have become available

### License server

The tools, applications, and libraries needed for the license server manager are part of the distribution and packaged in a separate subdirectory. The directory is named `installationdir/FlexLM`.

## 4.2 The License Server Manager

The *license server manager*, `lmgrd`, is one of two FLEXnet Licensing components that make up a license server system (the other being the vendor daemon). It handles the initial contact with a Delft-Tyre application, passing the connection on to the appropriate vendor daemon. The purpose of the license server manager, `lmgrd`, is to:

- Start and maintain all the vendor daemons listed in the VENDOR lines of the license file.
- Refer application checkout (or other) requests to the correct vendor daemon, for example `madlic`.

A newer `lmgrd` can be used with an older vendor daemon, but a newer vendor daemon might not work properly with an older `lmgrd`. Always use the newest version of `lmgrd` as possible, which is available for download from Flexera.

This chapter provides procedural information on how to configure and manage the License Server Manager.

### lmgrd Command-Line Syntax

`lmgrd` is the main daemon for FLEXnet Licensing. When you invoke `lmgrd`, it looks for a license file that contains information about vendors and features and starts those vendor daemons.

## 4.2.1 Starting the License Server Manager on UNIX Platforms

The license server manager, and hence the license server system, must be started before the Delft-Tyre application can be used.

The license server manager, `lmgrd`, is started either manually on the command line or automatically at system startup. Both methods are discussed in the following sections.

***Note:** Start `lmgrd` only on the server machine specified on the SERVER line in the license file. If you are running three-server redundant license server systems, maintain an identical copy of the license file (as well as the `lmgrd` and the vendor daemons binaries) locally on each server machine rather than on a file server. If you do not do this, you loose all the advantages of having redundant servers, since the file server holding these files becomes a single point of failure.*

### 4.2.1.1 Manual Start

Start `lmgrd` from the UNIX command line using the following syntax:
`lmgrd -c license_file_list -L [+]debug_log_path`
where `license_file_list` is one or more of the following:

- the full path to a single license file
- a directory, where all files named `*.lic` in that directory are used
- `debug_log_path` is the full path to the debug log file

Prepending `debug_log_path` with the + character appends logging entries.

Start `lmgrd` by a user other than root since processes started by root can introduce security risks. If `lmgrd` must be started by the root user, use the `su` command to run `lmgrd` as a nonprivileged user:
`su username -c "lmgrd -c license_file_list -l debug_log_path"`
where *username* is a non-privileged user. You must ensure that the vendor daemons listed in the license file have execute permissions for *username*. The paths to all the vendor daemons in the license file are listed on each VENDOR line.

### 4.2.1.2 Automatic Start

On UNIX, edit the appropriate boot script, which may be `/etc/rc.boot, /etc/rc.local, /etc/rc2.d/Sxxx, /sbin/rc2.d/Sxxxx`. Include commands similar to the following. See the following notes for a full explanation.

```
/bin/su daniel -c 'echo starting lmgrd > \
/home/flexlm/v5.12/hp700_u9/boot.log'

/bin/nohup /bin/su daniel -c 'umask 022; \
/home/flexlm/v5.12/hp700_u9/lmgrd -c \
/home/flexlm/v5.12/hp700_u9/license.dat >> \
/home/flexlm/v5.12/hp700_u9/boot.log'

/bin/su daniel -c 'echo sleep 5 >> \
/home/flexlm/v5.12/hp700_u9/boot.log'

/bin/sleep 5

/bin/su daniel -c 'echo lmdiag >>\
/home/flexlm/v5.12/hp700_u9/boot.log'
```

```
/bin/su daniel -c '/home/flexlm/v5.12/hp700_u9/lmdiag -n -c\
/home/flexlm/v5.12/hp700_u9/license.dat >> \
/home/flexlm/v5.12/hp700_u9/boot.log'

/bin/su daniel -c 'echo exiting >>\
/home/flexlm/v5.12/hp700_u9/boot.log'
```

Please note the following about how this script was written:

- All paths are specified in full because no paths are assumed at boot time.
- Because no paths are assumed, the vendor daemon must be in the same directory as `lmgrd`, or the VENDOR lines in the license file must be edited to include the full path to the vendor daemon.
- The `su` command is used to run `lmgrd` as a non-root user, **daniel**. It is recommended that `lmgrd` not be run as root since it is a security risk to run any program as root that does not require root permissions. `lmgrd` does not require root permissions.
- **daniel** has a `csh` login, so all commands executed as **daniel** must be in `csh` syntax. All commands not executed as **daniel** must be in `/bin/sh` syntax since that is what is used by the boot scripts.

***Note:*** *This does not start the daemon until you reboot your license server machine.*

### 4.2.2 Starting the License Server Manager on Windows

The license server manager, and hence the license server system, must be started before the Delft-Tyre application can be used.

The license server manager, `lmgrd`, is started either manually on the command line or automatically at system startup. Both methods are discussed in the following sections.

***Note:*** *Start `lmgrd` only on the server machine specified on the SERVER line in the license file. If you are running three-server redundant license server systems, maintain an identical copy of the license file (as well as the `lmgrd` and the vendor daemons binaries) locally on each server machine rather than on a file server. If you do not do this, you loose all the advantages of having redundant servers, since the file server holding these files becomes a single point of failure.*

#### 4.2.2.1 Manual Start

It is not uncommon for the License Server Manager to be started on a Windows platform. This section provides procedural information on manual starts from the command line and how to configure the License Server Manager as a service.
To start `lmgrd` from the command line:

1. Start `lmgrd` as an application from a Windows command shell using the following syntax:
   ```
   C:\TNO Delft-Tyre\MF-Tyre MF-Swift 6.2.0.4\FlexLM> lmgrd -c license_file_list -L
   [+]debug_log_path
   ```
   where
   - `license_file_list` is one or more of the following:
     - the full path to a single license file, for example mfswift.lic
     - a directory, where all files named *.lic in that directory are used
   - `debug_log_path` is the full path to the debug log file

     Prepending `debug_log_path` with the + character appends logging entries.

Spaces in pathnames require double quotes around the path.
On Windows, `lmgrd` can be installed as a service to allow it to be started and stopped through a user interface and run in the background.

### 4.2.2.2 Automatic Start

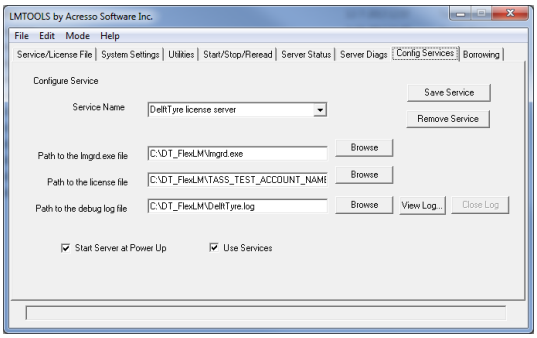
To configure a license server system such that it automatically starts, the license server has to be configured as a service, you must have Administrator privileges for this.

To configure a license server system as a service a graphical user interface to the license server manager tools is provided called LMTOOLS. Some of the functions LMTOOLS performs include:

- starting, stopping, and configuring FLEXnet license server systems
- getting system information, including hostids
- getting server status

In order to control the operation of `lmgrd`, and hence the license server system, from the LMTOOLS user interface, you first must configure it as a license server manager service like:

1. Run the file *lmtools.exe* which can be found in the folder `installationdir\FlexLM`.
2. Click the **Configuration using Services** button, and then click the **Config Services** tab.
3. In the **Service Name**, type the name of the service that you want to define, for example, **Delft-Tyre license server**.
4. In the **Path to the lmgrd.exe file** field, enter or browse to lmgrd.exe for this license server system.
5. In the **Path to the license file** field, enter or browse to the license file for this license server system.
   Note: The licenses are provided via the local sales representative.
6. In the **Path to the debug log file**, enter or browse to the debug log file that this license server system writes. Prepending the debug log file name with the + character appends logging entries. The default location for the debug log file is the `c:\winnt\System32` folder. To specify a different location, make sure you specify a fully qualified path. Note that the log file is not automatically created. It needs to be created by hand.
7. In order for the license server system to start up automatically at system start-up time:
   7.1. Make this license server manager a Windows service by selecting the **Use Services** check box (otherwise, it becomes a FLEXnet Licensing service).
   7.2. Configure it to start at system startup time by selecting the **Start Server at Power Up** check box.
   From now on, when the machine is rebooted, this license server manager starts automatically as a Windows service

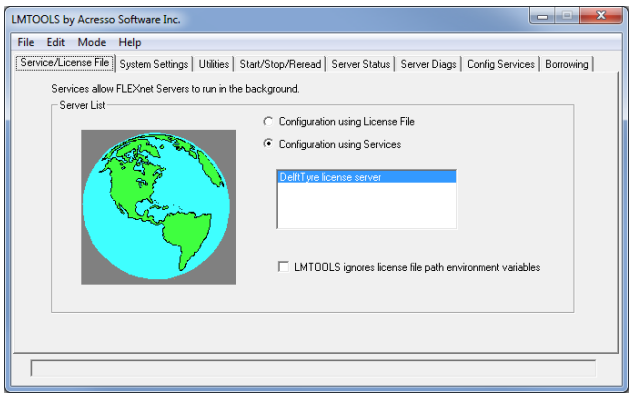



8. To save the new **Delft-Tyre license server** service, click **Save Service**.

Once the license server manager service is configured, `lmgrd` is started by starting the service from the LMTOOLS interface.

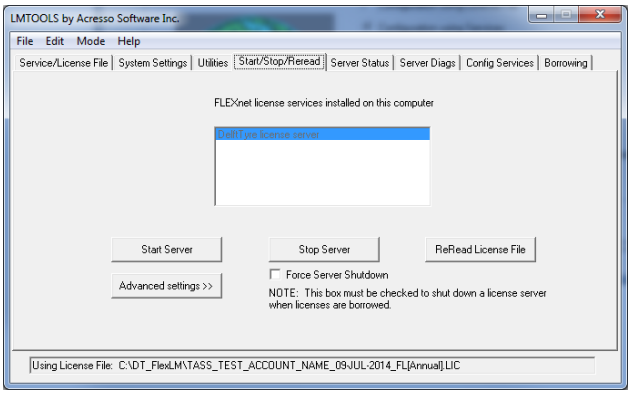
To start the service from the LMTOOLS interface:

1. Click **Configuration using Services** button.




2. Select the service name from the list presented in the selection box. In this example, the service name is **DelftTyre license server**.

3. Click the **Start/Stop/Reread** tab.




5. Start **DelftTyre license server** by clicking the **Start Server** button.

The **DelftTyre license server** system starts and writes its debug log output to the defined logfile.

## 4.3 License Troubleshooting Guide

1. Make sure that your FLEXlm license server is the latest available version

2. The environment variable MADLIC_LICENSE_FILE should be set to “<portnumber>@<hostname>; portnumber is the connection port number of the license server, where hostname is the name of the license server without the domain name. See the first line in the license file for these details Note that the first hostname should be <portnumber>@localhost. This will enforce to look locally on your system if the machine is detached from the network.

3. Considerable delays in start up of the applications have been noticed if the license file contains license strings of which the end date has expired.

4. Considerable delays in start up of the applications have been noticed if nonexistent servers are assigned to the MADLIC_LICENSE_FILE environment variables or even in the registry.

5. The questions below have been taken from the FLEXlm user guide and are important when you have questions for Delft-Tyre Support:

   - What kind of machine is your license server running on?
   - What version of the operating system?
   - What machine and operating system is the application running on?
   - What version of FLEXlm does the FLEXlm-licensed application use? Use the lmver script, or, on Linux, execute the following command on your lmgrd, vendor daemon, and application: strings binary_name | grep

copy. Alternatively, lmgrd -v gives the lmgrd version, and this works with the vendor daemon also.
- What error or warning messages appear in the log file?
- Did the server start correctly? Look for a message such as: server xyz started for: feature1 feature2.
- What is the output from running lmutil lmstat -a?
- What is the output from running serveractutil –view?
- Are you running other products which are also licensed by FLEXlm?
- Are you using a combined license file or separate license files?
- Are you using a three-server redundant license server (multiple SERVER lines in your license file)?

6. Set the environment variable FLEXLM_TIMEOUT to 10000000. Its value is in microseconds and corresponds to 10 seconds. As the name suggests it sets a timeout value of 10 seconds to contact the FLEXlm server. This should be considered to be a last resort in case of failing license checkouts.

# Index