

Installation and Maintenance Guide

Version 10.0

Linux & Unix



Contents

	Page
1	IMPORTANT CHANGES FOR VERSION 10.0 4
1.1	Licensing 4
1.2	Changes to Installation Organisation 4
2	INSTALLATION ORGANISATION 5
2.1	Version 10.0 Installation Organisation 5
2.2	Pre Installation 9
2.3	Single User / Single Machine Installation 10
2.4	Multiple Machine Network Installation 11
2.5	Multiple Machine Local Installation 12
2.6	Making the oasys_10 shell accessible to users 12
3	CUSTOMISING THE OASYS SHELL 14
3.1	Environment Variables for licensing 16
3.2	Other Environment Variables 17
4	SETTING UP USER PREFERENCES 19
4.1	The 'oa_pref' file 19
4.2	The preferences editor 21
4.3	Locking Preferences 23
5	AUTOMATIC LICENSE TIMEOUTS 24
6	CUSTOMISING THE SHELL FOR SUBMITTING LS-DYNA JOBS 26
7	TUNING THE GRAPHICS DRIVER 27
7.1	Finding out what graphics card and driver you have installed.27
7.2	Tuning an NVidia graphics driver 27
7.3	Tuning an AMD / ATI driver 28
8	THE FLEXLM LICENSING SYSTEM 29
8.1	Installing FLEXlm 29
8.2	Obtaining Oasys FLEXlm Licenses 29
8.3	Entering Oasys FLEXlm licenses 29
8.4	Starting the Oasys FLEXlm license daemon 29
8.5	Redundant License Servers 30

Intended audience

This document is written for the System Manager responsible for loading and maintaining the Oasys Ltd LS-DYNA 10.0 Environment software. No special (engineering) knowledge of the software is required.

Further information may be obtained from:

UK Arup Group Ltd
The Arup Campus
Blythe Gate
Blythe Valley Park
Solihull, West Midlands
B90 8AE
United Kingdom
Tel: +44 (0) 121 213 3399
Fax: +44 (0) 121 213 3302
Email: dyna.support@arup.com
Web: www.oasys-software.com/dyna

China Arup
39/F-41/F
Huai Hai Plaza
Huai Hai Road (M)
Shanghai
China 200031
Tel: +86 21 6126 2875
Fax: +86 21 6126 2882
Email: china.support@arup.com
Web: www.oasys-software.com/dyna

India Arup
Plot 39, Ananth Info Park
Opp. Oracle Campus
HiTec City
Madhapur Phase II
Hyderabad 500081
India
Tel: +91 40 4436 9797/98
Email: india.support@arup.com
Web: www.oasys-software.com/dyna

or contact your local Oasys Ltd distributor

1 IMPORTANT CHANGES FOR VERSION 10.0

1.1 Licensing

The version 10.0 software has been compiled using version 11.8 of FLEXlm. Previous releases used FLEXlm 10.8 so you **MUST** update your FLEXlm license servers to version 11.8.

The version 10.0 license file and license daemons are backwards compatible with the 9.1, 9.2, 9.3 and 9.4 software releases. This means that any existing 9.1, 9.2, 9.3, 9.4 software will continue to work with the new license files and servers

However, to run 64 bit versions of 9.3.1 (and earlier) you will need to replace the file flcheck93.exe in the 9.3 installation directory with a new version of flcheck93.exe provided with (but not used by) the 10.0 installation.

If this is done, the 9.3 software will work with the new license file format.

1.2 Changes to Installation Organisation

The installation directory structure was changed in version 9.4 to make it easier to manage and maintain the software on large networks. Version 10.0 uses the same directory structure as 9.4.

2 INSTALLATION ORGANISATION

2.1 Version 10.0 Installation Organisation

In version 10.0 the option is provided to separate a top-level “administration” directory from the “installation” one where the executables are located.

For large installations on many machines this allows central configuration and administration files to exist in one place only, but executables to be installed locally on users’ machines to give better performance.

Version 10.0 also allows for the following items to be configured

- The location for user manuals and other documentation.
- The definition of a user’s home directory.
- The definition of the temporary directory for scratch files.

In addition parsing of the “oa_pref” (preferences) file will now handle environment variables, so that a generic preference can be configured to give a user-specific result, and preferences may be “locked” so that those set at the administration level cannot be changed by users.

These changes are entirely optional, and users performing a simple installation on a single machine do not need to make any changes to their existing installation practice.

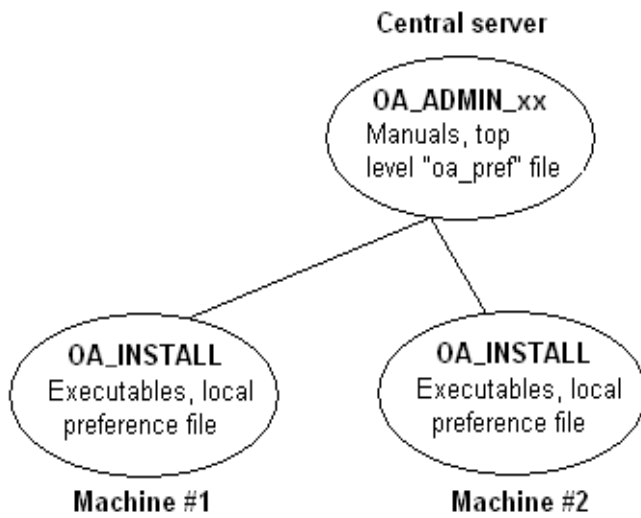
Here are some diagrams which illustrate how installation might be carried out in various different scenarios.

a) Single user installation on one machine



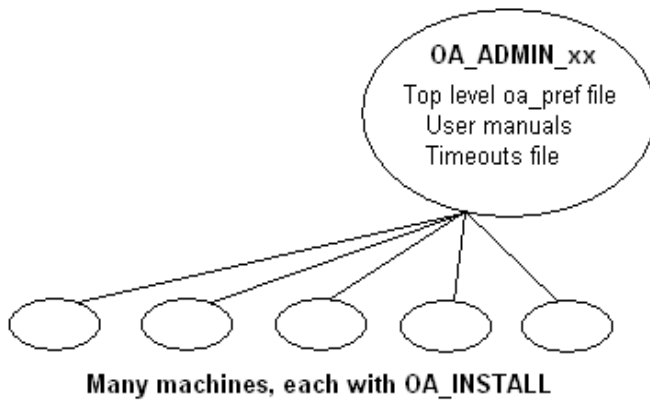
There is no need to worry about separating administration and installation directories, and the default installation of all files in and below the single installation directory will suffice.

b) A few machines on a small network, each user has his own machine



The top level administration directory can be installed on a network server, possibly also locating the manuals centrally. Each user's machine has its own "installation" directory to give good performance, but there is no need to manage home or temporary directories centrally since each user "owns" his machine. If network performance is good an alternative would be to install executables on the central server, meaning that local OA_INSTALL directories are not required.

c) Large corporate network



The top level oa_pref file will set the "home" directory for users so that their home directory is the same regardless of the machine they use. Timing out of idle licenses is managed centrally via the "timeouts" file. Corporate policies can also be enforced if required by "locking" preferences in the top level oa_pref file. Depending on network size & performance executables and manuals could be located on each machine, or on local server hubs, or centrally.

These configurations are not mandatory and are simply examples, you should choose the one that suits your needs.

Note the "_xx" in OA_ADMIN_xx and OA_INSTALL_xx refers to a version number, ie OA_ADMIN_10 for this release 10.0. This suffix is not required, but it is recommended as it will provide an easy way of organising parallel installations of future releases on a single machine.

Note also that while the various directories (OA_ADMIN_xx, OA_INSTALL_xx, OA_MANUALS, etc) can be defined by environment variables this is not recommended because it is inflexible, and also it would not permit two different installations to have different directories on the same machine.

It is recommended that the options in the top level `oa_pref` file to define these directories are used instead, since this encapsulates the definitions in a single place, permits multiple installations to co-exist on the same machine, and makes administration easier. See “`oa_pref` file options” in the table below.

2.1.1 Details of directory names

Directory	Status	Directory Content and purpose	oa_pref file option
OA_ADMIN_XX <i>or</i> OA_ADMIN	<i>Optional</i>	Top level configuration files. (XX = 10 for release 10.0, thus OA_ADMIN_10) Admin level <code>oa_pref</code> file Other configuration files Timeout configuration file The generic version of this name, OA_ADMIN will be searched for if no release-specific version is set.	
OA_INSTALL_XX <i>or</i> OA_INSTALL	Required	All executables Installation level <code>oa_pref</code> file	oasys*install_dir
OA_MANUALS	<i>Optional</i>	Specific directory for user manuals. If not defined then will search in: OA_ADMIN_XX/manuals (xx = major version number) OA_INSTALL/manuals	oasys*manuals_dir
OA_HOME	<i>Optional</i>	Specific "home" directory for user. If not defined will use: \$HOME (Unix/Linux) %USERPROFILE% (Windows)	oasys*home_dir
OA_TEMP	<i>Optional</i>	Specific "temporary" directory for user. If not defined will use: P_tmpdir (Unix/Linux) %TEMP% (Windows)	oasys*temp_dir

OA_INSTALL_xx

Previously the software used the OA_INSTALL (renamed from OASYS) environment variable to locate the directory the software was installed in.

On UNIX/LINUX systems the "oasys_10" script that starts the SHELL automatically sets this Environment Variable and passes it to any application started from the SHELL. If you run applications directly from the command line and bypass the SHELL then you should set OA_INSTALL so that the software can locate manuals and other required files.

OA_ADMIN_xx

Users wishing to separate configuration and installation directories will be able to do so by making use of the new top level OA_ADMIN_xx directory (OA_ADMIN_10 for release 10.0).

If the OA_ADMIN_xx directory is used it will be necessary to set up an environment variable of this name to refer to it, however this should normally be the only environment variable required in the whole installation.

2.1.2 Dynamic configuration using the top level oa_pref file

While all the OA_... directories may be specified by environment variables of the same name it is recommended that you do not do this, but instead use the facility to set non-standard directory names dynamically using preferences in the top level oa_pref file.

For example:

Release 10.0	Release 10.1
Top level directory OA_ADMIN_10	Top level directory OA_ADMIN_101
oa_pref file in OA_ADMIN_10 contains: oasys*install_dir: <pathname for 10.0 installation> oasys*manuals_dir: <pathname for 10.0 manuals> oasys*home_dir: <pathname for home directory> oasys*temp_dir: <pathname for temporary files>	oa_pref file in OA_ADMIN_101 contains: oasys*install_dir: <pathname for 10.1 installation> oasys*manuals_dir: <pathname for 10.1 manuals> } would almost certainly be unchanged between } major versions, although they could be different } if desired

For example if the Oasys 10.0 software has been installed in /home/oa10 , then:

```
oasys*install_dir: /home/oa10
```

will enable all users' installations to find their locally stored executables.

Pathnames using environment variables will be deconstructed during oa_pref file reading, and this can be exploited to set user-specific paths using a generic definition. For example:

oasys*home_dir: \$HOME/oa10

Would set a home directory.

2.1.3 The hierarchy of oa_pref file reading

The oa_pref preference file contains code-specific preferences that can be used to modify the software behaviour.

This file can be located in multiple locations which are searched in following order:

OA_ADMIN_xx / OA_ADMIN	Top level configuration
OA_INSTALL_xx / OA_INSTALL	Installation level
OA_HOME	User's personal "home" file
Current working directory	File specific to the current directory (rarely used)

The rules for reading these files are:

If a given directory does not exist, or no file is found in that directory, then no action is taken. This is not an error.

A more recently read definition supersedes one read earlier, therefore "local" definitions can supersede "global" ones (unless they are locked).

If two or more of the directories in the table above are the same, then that file is only read once from the first instance.

More information about preferences, including the ability to "lock" them, is given in section 4.

2.1.4 Protection and ownership of installation directories.

Oasys software does not require Administrator / Root privileges for installation.

It is recommended, but not required, that OA_ADMIN_xx and OA_INSTALL_xx directories be protected "read and execute only" for unprivileged users. If top level preferences are to be locked or idle time-outs configured, then write protection will be required to prevent users from subverting these settings.

2.2 Pre Installation

The UNIX/LINUX installation files can be downloaded from the following website.

http://www.oasys-software.com/dyna/en/downloads/oasys_software.shtml

All of the installation files are supplied as a single gzipped TAR file.

Separate downloads are available for each supported operating system. The LINUX installation packages contain both 32 and 64 bit versions of D3PLOT, T/HIS and PRIMER while the UNIX ones only contain the 64 bit versions.

If you are going to install a license server on a different platform to that which you are running the software on then you can download just the FLEXlm tools for the machine type you want to use as the license server.

After downloading the installation file you should copy it to a directory where you can uncompress it and then untar it.

```
gunzip install_10.tar.gz
```

```
tar xvf install_10.tar
```

This should produce the following 5 files

- peripheral.tar
- environment.tar
- manuals.tar
- flex.tar
- setup.csh

As part of the installation process you will need to select the license type. If you are using a license server you will need to know the name of the license server (and the port number if this has been set). If you are using a node locked license then you will need to know the full pathname of where the license file is located.

If you are going to set up a separate OA_ADMIN administration directory then you should also know the pathname for the directory that you want to use as the installer script will prompt you for it.

2.3 Single User / Single Machine Installation

If you are installing the software on a single machine it is recommended that the software is installed on a local disk.

2.3.1 Installation

- To start the installation process run setup.csh from the command line
- When prompted enter the full pathname of the directory that you want to install the software in and the license information. If you do not know the name of the license server or if you have not yet obtained a node locked license file then you can enter any machine name / filename and change them later (see section 3).
- After entering the license information the installation process will continue and you will be offered the opportunity to set up an OA_ADMIN_xx directory (see section 2.1.1). For a single user / single machine installation this can be skipped.
- The installer should then complete the installation without prompting for any more information.

2.3.2 Post Installation

- a) The Oasys software is designed to be run via the command script “oasys_10”. As part of the installation process a number of Environment Variables will have been automatically set up within this script. See section 3 for more details on these Environment Variables and how to modify them.
- b) Configure the preference file “oa_pref”, (see section 5)
- c) Optional - Configure the SHELL for submitting LS-DYNA analysis (see section 7).

2.3.3 Making the oasys_10 shell accessible to users

The version 10.0 software is designed to be accessed via the command “oasys_10”. For more information on how to make this command available to all users see section 2.6.

2.4 Multiple Machine Network Installation

Installing the software onto a network drive for access from multiple machines.

2.4.1 Installation

- a) To start the installation process run setup.csh from the command line
- b) When prompted enter the full pathname of the directory that you want to install the software in and the license information. If you do not know the name of the license server or if you have not yet obtained a node locked license file then you can enter any machine name / filename and correct them later (see section 3).
- c) After entering the license information the installation process will continue and you will be offered the opportunity to set up an OA_ADMIN directory (see section 2.2.1).
- d) The installer should then complete the installation without prompting for any more information.

2.4.2 Post Installation

- a) The software is designed to be run via the command script “oasys_10”. As part of the installation process a number of Environment Variables will have been automatically setup within this script. See section 3 for more details on these Environment Variables and how to modify them.
- b) Configure the preference file “oa_pref”, (see section 5)
- c) Optional - Configure the SHELL for submitting LS-DYNA analysis (see section 7).

2.4.3 Making the oasys_10 shell accessible to users

The version 10.0 software is designed to be accessed via the command “oasys_10”. For more information on how to make this command available to all users see section 2.6.

2.5 Multiple Machine Local Installation

If you are going to install the software on multiple machines then you can either follow the procedure outlined in section 2.4 for each machine or you can install the software once and then copy the installation to each machine.

2.5.1 Installation

- a) To start the installation process run `setup.csh` from the command line
- b) When prompted enter the full pathname of the directory that you want to install the software in and the license information. If you do not know the name of the license server or if you have not yet obtained a node locked license file then you can enter any machine name / filename and correct them later (see section 3).
- c) After entering the license information the installation process will continue and you will be offered the opportunity to set up an `OA_ADMIN` directory (see section 2.1.1).
- d) The installer should then complete the installation without prompting for any more information.

2.5.2 Post Installation

- a) The software is designed to be run via the command script “`oasys_10`”. As part of the installation process a number of Environment Variables will have been automatically setup within this script. See section 3 for more details on these Environment Variables and how to modify them.
- b) Configure the preference file “`oa_pref`”, (see section 5)
- c) Optional - Configure the SHELL for submitting LS-DYNA analysis (see section 7).

After you have configured the preference file the complete installation directory can be copied to the other machines.

2.5.3 Making the `oasys_10` shell accessible to users

The version 10.0 software is designed to be accessed via the command “`oasys_10`”. For more information on how to make this command available to all users see section 2.6.

2.6 Making the `oasys_10` shell accessible to users

The Oasys software is designed to be run via the command script “`oasys_10`”. This command has to be set so that all users can run the software with a single command. The way you go about this is up to you: the following are simply suggestions, you may prefer other methods.

(1) Set up a symbolic link in a directory accessed by all users.

On most systems each user's path is set up at login time to include directories that contain generally used commands and files. For example many systems include the `/usr/local/bin` directory. Putting a

symbolic link in such a directory that points to the `oasys_10` shell is a straightforward way to provide access to it for all users. You do this by going to that directory and typing the command:

```
ln -s pathname/oa10/oasys_10 oasys_10
```

Where *pathname* is the directory of `OA_INSTALL_xx`

(2) Set up an "alias" for selected users.

You may wish to restrict access to the software to a sub-set of users, in which case a more elegant solution might be to add an "alias" to their individual ".cshrc" files. (This assumes that they are running C shell.) You would do this by adding the line:

```
alias oasys_10 'pathname/oa10/oasys_10'
```

to their ".cshrc" files.

3 CUSTOMISING THE OASYS SHELL

On UNIX/LINUX systems the Oasys SHELL is accessed via the "oasys_10" command. If you have installed the software using the install script then the oasys_10 command file (in the /executables directory) will automatically be updated to contain the correct path for the directory the software was installed in and the name of the license server.

In addition to specifying the installation directory and the license server the "oasys_10" command file can also be set a number of other options on UNIX/LINUX systems.

After installing the software the "oasys_10" command file should contain the following. (This assumes the software was loaded in a directory called /prg and that the license server in a machine called atghps50.

```
#!/bin/csh -f
  onintr start_flush
#
# Set OA_INSTALL to point to the directory containing the software
#
  setenv OA_INSTALL "/prg/oa10/executables"
#
# Set OA_ADMIN directory
#
  setenv OA_ADMIN
#
# Set OASYS_LICENSE_FILE to either the Oasys flexlm license file or
# the license server
#
# e.g  setenv OASYS_LICENSE_FILE $OA_INSTALL/oasys_flexlm.dat
#      or setenv OASYS_LICENSE_FILE @hostname
#
  setenv OASYS_LICENSE_FILE @atghps50
#
# Set LSTC_FILE to either the lstc license file or the server
#
# e.g. setenv LSTC_FILE $OA_INSTALL/LSTC_FILE
#
# e.g. setenv LSTC_LICENSE_SERVER hostname
#       setenv LSTC_NOCLIENT      ON
#       setenv LSTC_LICENSE       network
#
  setenv LSTC_FILE $OA_INSTALL/LSTC_FILE
#
# Other environment variables
#
  setenv DISPLAY_FACTOR  automtaic
  setenv USERID          `whoami`
  setenv LICENSE_TYPE    "flexlm"
  set noglob
```

```
#
#
# Set EDITOR (if not set) to the command to invoke an external editor.
This
# EDITOR is currently used for editing comment lines in Oasys PRIMER.
#
# if(! $?EDITOR) then
# setenv EDITOR /usr/bin/kedit # LINUX
# setenv EDITOR /usr/dt/bin/dtpad # UNIX (CDE)
# endif
#
# Environment variables for post processors
#
# If FILE_SKIP has not been set then set it to 5
#
#   if(! $?FILESKIP) then
#       setenv FILE_SKIP 5
#   endif
#
# MACHINE SPECIFIC LINES
# =====
#
# extra line needed for some IBM machines
# setenv LANG En_US
#
# Extra line needed for SUN Solaris Machines
# setenv LD_LIBRARY_PATH /usr/openwin/lib
#
#
# Now start the main shell executable
#
#   set cwd = `pwd`
#   setenv PWD $cwd
#   $OA_INSTALL/xshell_10 $*
#
# exit
#
exit:
    exit
```

3.1 Environment Variables for licensing

3.1.1 `setenv OASYS_LICENSE_FILE / LM_LICENSE_FILE` <option>

Either `OASYS_LICENSE_FILE` or `LM_LICENSE_FILE` can be used to locate a valid license for the Oasys software. It is recommended that `OASYS_LICENSE_FILE` is used as this can speed up the checkout of licenses on systems where `LM_LICENSE_FILE` is used to find other license servers as well.

3.1.1.1 Floating Network License

If the software will be run using a license server this variable should be set to point to the license server machine using the machine's hostname:

```
setenv OASYS_LICENSE_FILE hostname
```

or if a non-default port has been specified for the license server:

```
setenv OASYS_LICENSE_FILE port@hostname
```

If you are using a triad license server then

you should specify all 3 license servers:

```
setenv OASYS_LICENSE_FILE port@host1:port@host2:port@host3
```

3.1.1.2 Fixed Stand-alone (nodelocked) Licenses

If the software will be using a node locked license file this variable should be set to point to the location of the license file:

```
setenv OASYS_LICENSE_FILE <INSTALL_DIR>\oasys_flexlm.dat
```

3.1.1.3 Multiple FLEXlm license files

If other software programs on the system are using FLEXlm this variable may already be set to point to a license file or a license server. If this variable is already set the Oasys license server or file can be added by specifying a list of values separated by semi-colons, for example:

```
setenv OASYS_LICENSE_FILE port@hostname1;@hostname2
```

3.1.2 `setenv LSTC_FILE / LSTC_LICENSE_SERVER` <option>

On UNIX/LINUX machines LS-DYNA can use either a nodelocked license or a floating license system.

3.1.2.1 Nodelocked License

If you are using a nodelocked license then LSTC_FILE should be set to the full pathname of the license file. By default this file should be called 'LSTC_FILE' and it should be located in the 'executables' directory.

```
setenv LSTC_FILE<INSTALL_DIR>/LSTC_FILE
```

3.1.2.2 Floating Network License

If you are using the floating license system then the variables LSTC_LICENSE_SERVER, LSTC_NOCLIENT and LSTC_LICENSE should be set as follows:

```
setenv LSTC_LICENSE_SERVER hostname (of license server)
```

```
setenv LSTC_NOCLIENT ON
```

```
setenv LSTC_LICENSE network
```

If you are using a triad license server then you should specify all 3 license servers:

```
setenv LSTC_LICENSE_SERVER '(host1 host2 host3)'
```

3.2 Other Environment Variables

The following parameters will require modifying in accordance with your system layout

```
setenv OA_INSTALL / OA_INSTALL_10<option>
```

```
setenv OA_ADMIN / OA_ADMIN_10 <option>
```

```
setenv MENU_AUTO_CONFIRM <option> (optional)
```

```
setenv FILE_EXIST_ACTION <option> (optional)
```

```
setenv ECHO_PREFERENCE <option> (optional)
```

3.2.1 setenv OA_INSTALL / OA_INSTALL_10 <option>

The Oasys shell variable must be modified so that it references the directory in which the shell is loaded. i.e if the software has been loaded in:

```
/prg/oa10/executables
```

Then this line should be set to:

```
setenv OA_INSTALL "/prg/oa10/executables"
```

This line should be set automatically by the installation script.

3.2.2 `setenv OA_ADMIN / OA_ADMIN_10 <option>`

If a top level administration directory is to be used then `OA_ADMIN_10` (for release 10.0) must be defined on all machines on which the software is to be run. This variable should be set to the full pathname of the administration directory.

3.2.3 `setenv MENU_AUTO_CONFIRM <option>`

This variable is often used when replaying command files which, when recorded, paused and asked the user to confirm things. (For example `HELP` and Warning messages.) Possible options for this variable are, `true` and `false`.

If the variable is set (`true`) then these will not pause and will behave as if the user had pressed "OK" - meaning that command files can play back without user intervention. As a general rule this variable should not be set for interactive usage, but may be required when performing "batch" type operations.

3.2.4 `setenv FILE_EXIST_ACTION <option>`

This variable controls the action to be taken when opening a file for output, and the file already exists. Possible options for this variable are "none", `overwrite` and `append`.

Normally you will be prompted for the action to be taken when a file selected for output already exists. However if this variable is set to `overwrite` or `append` then the relevant action will be taken automatically.

This is generally used when playing automatic post-processing batch scripts, and should not be set for normal interactive usage.

3.2.5 `setenv ECHO_PREFERENCE <option>`

If this variable is set to "1" then any command line arguments used to start `T/HIS`, `PRIMER` or `D3PLOT` will be echoed to the screen along with any settings read from preference files.

4 SETTING UP USER PREFERENCES

4.1 The 'oa_pref' file

This file contains code-specific preferences that can be used to modify the behaviour of the software suite. It is optional, and where entries (or the whole file) are omitted programs will revert to their default settings.

4.1.1 'oa_pref' naming convention and locations

The preferences are stored in a file called "oa_pref". This file can exist in multiple locations which are searched in the following order:

The optional administration directory defined by the environmental variable (\$OA_ADMIN or \$OA_ADMIN_xx - where xx is the release number).

The site-wide installation directory defined by the environment variable (\$OA_INSTALL_xx)

The user's home directory \$OA_HOME which defaults to: \$HOME (Unix/Linux) or %USERPROFILE% (Windows)

The current working directory

(see Section 2 for an explanation of the directory structure).

All four files are read (if they exist) with the last preference read being the one used; this means the file can be customised for a particular job or user if necessary.

Files do not have to exist in any of these locations; if none exists the programme defaults will be used.

Typically the following should be set:

Organisation-wide options in the version in \$OA_ADMIN_xx and/or \$OA_INSTALL,

User-specific options in \$OA_HOME

Project-specific options in the current working directory.

4.1.2 File Format

The file contains preferences for:

- all the software (lines commencing oasys*)
- SHELL (lines commencing shell*)
- THIS (lines commencing this*)
- D3PLOT (lines commencing d3plot*)
- PRIMER (lines commencing primer*)
- REPORTER (lines commencing reporter*)

All lines take the format

<program name> * <preference name> : <preference value>.

The general copy of the preference file should be present in the \$OA_ADMIN_xx and/or \$OA_INSTALL_xx directory. This should contain the preferences most suitable for all software users on the system.

An individual's specific preferences file can be stored in the individual's home area or \$OA_HOME. This can be used to personally customise the software to the individual's needs.

Whenever a program with preferences in the oa_pref file is fired up, the program will take preferences in the following order:

1. from the general preference file in the \$OA_ADMIN_xx directory (if it exists)
2. then the \$OA_INSTALL_xx directory
3. then from the file in the user's home area (\$OA_HOME)
4. then from the current working directory

Preferences defined in the general oa_pref file can be superseded by preferences of the same name in the user's personal file but they can't be removed by it.

From version 9.4 onwards preferences can be locked. If a preference is locked it cannot be changed in an oa_pref file in a more junior directory. To lock a preference use the syntax '**program#**' rather than '**program***'.

An example of the file is shown below to illustrate the content of the file

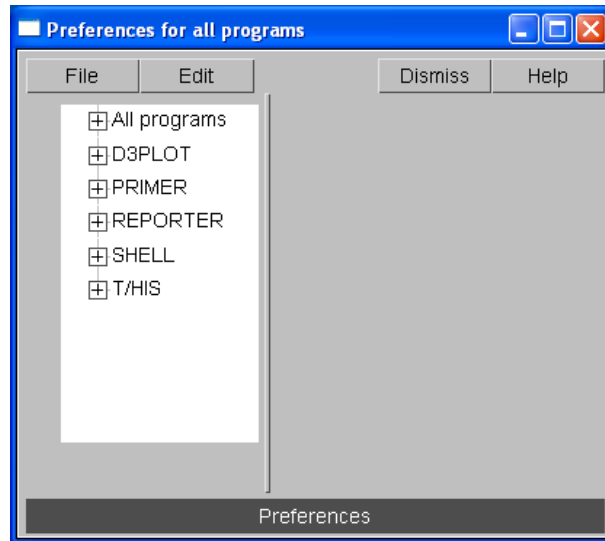
```
# Preferences file for software.
#
# Preferences for SHELL
shell*queue_cpu: 0
#
# Preferences for THIS
this*laser_paper_size: A4
#
# Preferences for D3PLOT
d3plot*overlay_colour: grey
#
# Preferences for PRIMER
primer*overlay_mode off
```

An example of a locked preference would be:

```
primer#background_colour: white
```

4.2 The preferences editor

The editor can be accessed from within the SHELL or from within D3PLOT, T/HIS, PRIMER or REPORTER. The preference settings for each program are listed in the appropriate manual.



4.2.1 The preferences editor layout

The preferences editor window is divided into two frames with a menu bar across the top.

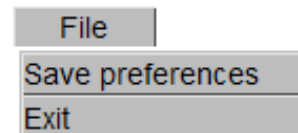
4.2.2 Menu Bar



File options:

Save preferences: Save current preference settings. This will save the personal oa_pref file in the user's home directory. Only those preferences which differ from the preferences saved in the general oa_pref file will be saved.

Exit: Exit the preferences editor without saving.



Edit options:

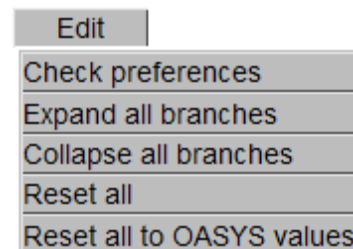
Check Preferences: Checks the current preferences for any errors. These errors will be listed in a separate window detailing the preferences with the errors and the nature of those errors

Expand all branches: Expands the categories in the Left hand frame.

Collapse all branches: Collapses the categories in the Left hand frame

Reset all: Resets all values.

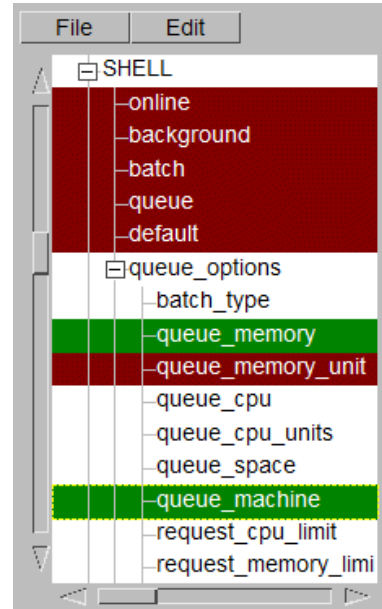
Reset all to OA_INSTALL values: Resets all values to the defaults stored in the main \$OA_INSTALL preference file



4.2.3 The preferences editor Left hand frame

The left hand frame will contain the names of all preferences available to set. Preferences will be listed under the headings: PRIMER, D3PLOT, T/HIS, REPORTER and SHELL according to which program they are applicable to.

These categories can be expanded to reveal their respective preferences/contracted to hide their preferences by clicking on the box to the left of the respective category, alternatively, use the edit drop down menu and select Expand all branches or collapse all branches.



- Green** Means that the option has been read from your \$HOME/\$USERPROFILE file.
- Red** Means that the option has been read from the \$OA_INSTALL file.
- Magenta** Means that the option had been read from the \$OA_ADMIN file.

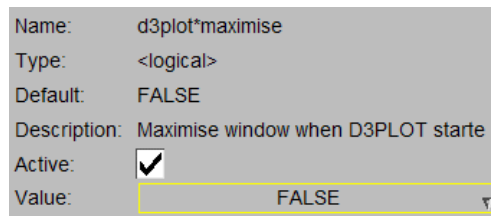
Preferences which aren't highlighted indicate preferences that haven't been set.

Preferences in **bold** type indicate preferences which haven't been assigned the default value.

A list of all the preferences available and their default value can be found in any oa_pref file written by the preferences editor.

4.2.4 The preferences editor Right hand frame

The right hand frame will contain information about the currently selected preference and provides the opportunity to edit this preference:



Name: States the name of the currently selected preference.

Type: Specifies the type of variable applicable to this preference.

Default: States the default value of the preference.

Description: Provides a brief description of the function performed by this preference.

Active tab: Highlighted in Green when the preference has been assigned a value. Press this tab to activate/ deactivate the currently selected preference. If the

currently selected preference was defined in the general oa_pref file, deselecting this will bring up an error message as it is not possible to deselect preferences stored in the general oa_pref file.

Value: States the currently selected value for the preference. Clicking on the arrow to the left of this box brings up a drop-down menu which lists the possible values this preference can take and allows the user to select one of these values.

4.3 Locking Preferences

From version 9.4 onwards preferences can be locked. Beside each option in the preference editor is a padlock symbol. If the symbol is green then the option is unlocked, if it is red then it is locked. If a preference option has been locked in a file that the user can not modify then an error message will be generated if the user tries to edit that option.

If a user manually edits the "oa_pref" file to try and set an option that has been locked in another preference file then the option will be ignored in the users preference file.

An unlocked preference is defined in the oa_pref file by:

```
<programme> * <preference> : <value>
```

A locked preference replaces the "*" with a "#", thus:

```
<programme> # <preference> : <value>
```

5 AUTOMATIC LICENSE TIMEOUTS

From version 9.4 onwards each application can be setup to exit automatically if it remains idle for a specified time. When the application exits it will automatically release any licenses that are being used and return them to the pool of free licenses.

The automatic license timeouts are controlled by a file called 'timeouts' located in the directory defined by the OA_ADMIN_XX or the OA_INSTALL_XX environment variable.

The format of this file is

<application name> <idle time> <grace period>

e.g

```
#  
primer      60  5  
d3plot      60  5  
this        60  5  
#
```

The times are defined in minutes. The idle time should be > 0 and the grace period should be >=0.

Any line in the file starting with '#', '%' or '\$' is counted as a comment line.

Blank lines are ignored

Input is not case-sensitive

Input is free format, but each programme's settings must be on a single line.

When the idle time is exceeded a warning message will be displayed within the application's master window. This message will be displayed for the grace period specified; the application will be terminated if no response is detected. Giving a response resets both <idle> and <grace> counters so that a further <idle time> must elapse before a further warning is issued.

If programs terminate due to a license timeout the following occurs:

```
PRIMER: a copy of any models currently loaded will be saved in the users  
        home area or $OA_HOME  
D3PLOT: will just exit  
T/HIS:  will just exit
```

In all cases, the controlling terminal window receives a message explaining what has happened and why; this window will remain on both Windows and Unix/Linux platforms.

The warning notice is displayed within the master window of the application, not on the desktop. This is intentional to prevent users starting the application to grab a license then iconising/minimising it until they need it as the warning message will not be seen if the application is minimised.

Note that the timeouts file must be write-protected against users either by protecting the file, or the directory in which it exists, otherwise users will be able to change the file content.

6 CUSTOMISING THE SHELL FOR SUBMITTING LS-DYNA JOBS

In addition to accessing the Oasys software, the SHELL can be used to submit LS-DYNA jobs. To use the SHELL to submit LS-DYNA the following should be configured:

1. General Submission Options
2. The versions of LS-DYNA available
3. Queuing Options
4. MPI commands for MPP submission

For more details on how to perform all of these please see the Oasys SHELL manual.

7 TUNING THE GRAPHICS DRIVER

Oasys software makes intensive use of 3D graphics, and experience has shown that the default settings in the drivers of commonly used graphics cards can cause problems: typically visual artefacts and/or “stuttery” animation performance.

The following section gives suggested tuning setting for commonly used cards, but if you experience problems and your card is not listed below please contact Oasys for help and advice.

7.1 Finding out what graphics card and driver you have installed.

Unfortunately there is no ready-made tool on Linux/Unix to provide details of the graphics card installed, but the following procedure should work:

Type `glxinfo | grep -i string` which should give card manufacturer and name.

For example on a machine with an ATI card this might produce:

```
OpenGL vendor string: ATI Technologies inc
OpenGL renderer string: ATI FirePro V7750 (FireGL)
OpenGL version string: 3.3.10225 Compatibility Profile FireGL
```

And on a machine with an NVidia card:

```
OpenGL vendor string: NVIDIA Corporation
OpenGL rendered string: Quadro FX 3800/PCI/SSE2
OpenGL version string: 3.3.0 NVIDIA 256.35
```

Once you know the make of the card you can then look in file `/var/log/Xorg.0.log` for more information about its driver. The actual filename may vary with the version of Linux/Unix, but something like the following should work, the example here being for an NVidia card:

```
grep -i nvidia /var/log/Xorg.0.log | grep -i driver
```

Which should produce something like:

```
(II) Loading /usr/lib64/xorg/modules/drivers/nvidia_drv.so
(II) NVIDIA dlloader X driver 256.35 Wed Jun 16 18:45:02 PDT 2010
(II) NVIDIA Unified driver for all Supported NVIDIA GPUs
```

From which you can discern that the driver release is 256.35 dated 16th June 2010

It is recommended that if your graphics driver is significantly out of date that you consider upgrading it to a more recent version. This is not mandatory, and if the machine is working well there is a strong case for “if it isn’t broken, don’t fix it”; but certainly the first step to be taken if graphics problems arise is to upgrade an out of date driver.

7.2 Tuning an NVidia graphics driver

No tuning is required for NVidia drivers under Linux.

7.3 Tuning an AMD / ATI driver

On Linux ATI create the following XML file:

/etc/ati/atiogl.xml

In which configuration settings for individual applications are stored. This file needs to be edited to include settings for the Oasys Ltd software suite. Here is the top of a typical file:

```
<PROFILES>
<!-- ===== -->
<!-- Workstation Applications -->
<!-- ===== -->
<!-- 3ds max -->
  <_3dsmax>
    <OpenGLCaps>0x00008040</OpenGLCaps>
    <OpenGLCapsEx>0x00000000</OpenGLCapsEx>
    <CrossFireCaps>0x00000010</CrossFireCaps>
  </_3dsmax>
```

The following lines need to be added. They can go anywhere in the file, but you might wish to insert them in the correct alphabetical location:

```
<!-- D3PLOT -->
  <d3plot10>
    <OpenGLCaps>0x00008000</OpenGLCaps>
  </d3plot10>
  <d3plot10_64>
    <OpenGLCaps>0x00008000</OpenGLCaps>
  </d3plot10_64>
<!-- PRIMER -->
  <primer10>
    <OpenGLCaps>0x00008000</OpenGLCaps>
  </primer10>
  <primer10_64>
    <OpenGLCaps>0x00008000</OpenGLCaps>
  </primer10_64>
<!-- THIS -->
  <this10>
    <OpenGLCaps>0x00008000</OpenGLCaps>
  </this10>
  <this10_64>
    <OpenGLCaps>0x00008000</OpenGLCaps>
  </this10_64>
```

Once edited simply save the file in its current location.

8 THE FLEXlm LICENSING SYSTEM

The LS-DYNA Environment software (PRIMER, D3PLOT, T/HIS and REPORTER) use the FLEXlm licensing system.

8.1 Installing FLEXlm

All of the files needed to install FLEXlm on the system will have been copied into the installation directory when the software was installed. The Oasys software is designed to be used with either 'floating' or 'node locked' licenses.

8.2 Obtaining Oasys FLEXlm Licenses

For floating (server) licenses: the hostname and FLEXlm host ID of the server machine only are required (although this does not have to be a machine that the Oasys software is installed on it is recommended to be).

For node locked licenses: the hostname and FLEXlm host ID of each machine upon which the software will run are required.

To generate this information go to:

OA_INSTALL_xx directory and run lmtools.

Please send the output generated by the **Hostid** button to Oasys Ltd by email: dyna.license@arup.com or by fax: +44 (0) 121 213 3302.

8.3 Entering Oasys FLEXlm licenses

The Oasys FLEXlm license codes are contained in a file called '**oasys_flexlm.dat**' which should be placed in the directory where the executables themselves have been installed (ie OA_INSTALL_xx). Typically Oasys Ltd will send license codes by e-mail.

8.3.1 Node locked license

On UNIX systems the installation script should automatically have set the environmental variable OASYS_LICENSE_FILE to point to file **oasys_flexlm.dat**, specifying its full pathname. If not, set it by editing the oasys_10 script.

8.3.2 Floating (server) license

If a floating (server) license is to be used the Oasys FLEXlm license daemon should be started.

8.4 Starting the Oasys FLEXlm license daemon

When the Oasys software is installed the '**start_flexlm**' script is automatically added to the OA_INSTALL_xx directory. As part of the installation process this file should have been automatically modified so that the environment variable

OASYS_LICENSE_FILE has been set to point to the full pathname of the 'oasys_flexlm.dat' license file (see section 3.1)

Further, the oasys_10 script should have

```
setenv OASYS_LICENSE_FILE @hostname
```

where hostname is the license server.

To start the Oasys FLEXlm license daemon go into the installation directory and enter:

```
./start_flexlm
```

If the command fails with the following error message

```
“License manager: can’t initialize: Cannot find license file”
```

check that the LM_LICENSE_FILE environment variable has been set correctly in 'start_flexlm'.

If the license manager starts successfully it is recommended that system boot sequence is modified to ensure the **start_flexlm** script file is automatically run whenever the system is rebooted.

8.5 Redundant License Servers

If planning to use redundant servers it is advisable to select stable systems as server machines; do not pick systems that are frequently rebooted or shut down. Redundant license server machines are any supported server machines.

FLEXlm supports two methods of redundancy:

- Via a license-file list in the LM_LICENSE_FILE environment variable
- Via a set of three redundant license servers

With LM_LICENSE_FILE list redundancy, each one of a group of license servers serves a subset of the total licenses. The end user sets LM_LICENSE_FILE to a list of license files, where each license file refers to one of the license servers. The application then tries each server in the list in order until it succeeds or gets to the end of the list.

With three-server redundancy, if any two of the three license servers are up and running (two out of three license servers is referred to as a quorum), the system is functional and serves its total complement of licenses.

8.5.1 Three-Server Redundancy

The machines that comprise a three-server redundant configuration should

- Run the same operating system
- Have excellent communications
- Reside on the same subnet

The three servers must be located physically close to each other. This form of redundancy requires that the servers exchange heartbeats periodically and poor

communications can cause poor performance. Avoid configuring redundant servers with slow communications or dial-up links.

Three-server redundancy is designed to provide hardware failure protection only and does not provide load-balancing (if load-balancing is desired use the `LM_LICENSE_FILE` list option).

With three-server redundancy only one of the three servers is “master,” capable of issuing licenses. If the “master” machine fails one of the remaining machines takes over the “master” role and continues to serve licenses. Since all clients must contact the “master” all clients must have reliable networking to a single machine.

Appendix A - FLEXlm License Administration Tools

A1 UNIX

All license administration tools are contained in the single executable `lmutil`. `lmutil` contains the following utility programs:

lmborrow
lmdiag
lmdown
lmhostid
lminstall
lmnewlog
lmpath
lmremove
lmreread
lmstat
lmswitch
lmswitchr
lmver

All of these utilities are accessed by the command

lmutil <lm_command>

All utilities take the following arguments: `-v` print version and exit

`-c license_file` operate on 'license file'

A2 LMBORROW

`lmborrow` supports borrowing of licenses that contain the `BORROW` attribute. It must be run on the machine where licenses are borrowed. It is used to perform the following:

- Initiating borrowing by setting the borrow period
- Clearing the borrow period
- Determining borrow status
- Returning a borrowed license early

A2.1 Initiating Borrowing

To initiate borrowing, the user sets the borrow period:

lmdiag {vendor | all} enddate [time]

vendor The vendor daemon name that serves the licenses to be borrowed, or all specifies all vendor daemons in that license server system.

enddate [time] Date the license is to be returned in dd-mmm-yyyy format. time is optional and is specified in 24-hour format (hh:mm) in the FLEXenabled application's local time. If time is unspecified, the checkout lasts until the end of the given end date.

For example: **Imborrow primer 20-aug-2001 13:00**

This has the effect of setting **LM_BORROW** with the borrow period in either the registry (Windows) or in `$HOME/.flexImborrow` (UNIX).

To borrow licenses for the desired vendor, on the same day and the same machine that the user runs `Imborrow`, run the application(s) to check out the license(s). If you run the application(s) more than once that day, no duplicate licenses are borrowed. No licenses are borrowed if the application is run on a day different than the date borrowing is initiated.

A2.2 Clearing the Borrowed License Setting

To clear the **LM_BORROW** setting in the registry or `$HOME/.flexImborrow`, issue the command:

Imborrow -clear

Clearing the **LM_BORROW** setting stops licenses from being borrowed until borrowing is initiated again. A user might run `Imborrow -clear` after he has borrowed licenses for features that are used offline if-before disconnecting from the network-he wants to run an application that checks out additional features, served by vendor, that are not meant to be borrowed. Clearing **LM_BORROW** does not change the status for already-borrowed licenses.

A2.3 Determining Borrowed License Status

To print information about borrowed features issue the following command on the machine from which they are borrowed:

Imborrow -status

The borrowing system does not have to be connected to the network to determine the status.

A2.4 Returning a Borrowed License Early

To return a borrowed license early, first reconnect the borrowing system back to the network then (from the same machine that initiated the borrowing) issue the command:

Imborrow -return [-fqdn][-c license_file_list] [-c display] feature

- fqdn** Directs `Imborrow` to access the borrowing system using its fully qualified host name. Use this option if the license was borrowed based on the fully qualified host name, rather than the relative distinguished name. Use `Imstat` to determine the format of the host name used when the license was borrowed.
- c license_file_list** Use the specified license file(s). In some configurations, the license file needs to be specified in order to return the license file early.

- d display** Used to specify the display from which the borrow was initiated. Required if your current display is different than what was used to initiate the borrow.
- On Windows, it is the system name or, in the case of a terminal server environment, the terminal server client name. On UNIX, it is in the form /dev/ttyxx or the X-Display name.
- feature** The name of the borrowed feature to be returned early. Use:

lmborrow -status

to get a list of borrowed feature names.

Returning the license early has the effect of clearing the **LM_BORROW** setting for the vendor daemon that serves the returned license.

If the borrowing system is not placed back on the network before attempting the early return, the license is not returned and **LM_BORROW** is kept intact. Additionally, an error message is issued to the end user with notification that the system needs to be connected to the network.

A3 LMDIAG

lmdiag allows you to diagnose problems when you cannot check out a license.

lmdiag [-c license_list] [-n] [feature[:keyword=value]]

- c license_list** path to the file(s) to diagnose. If more than one file, use colon separator of Unix, or semi-colon on PC.
- n** run in non-interactive mode; **lmdiag** will not prompt for any input in this mode. In this mode, extended connection diagnostics are not available.
- feature** diagnose this feature only.
- keyword=value** If a license file contains multiple lines for a particular feature, you can select a particular line for **lmdiag** to report on.

For example

lmdiag f1:HOSTID=12345678

will attempt a checkout on the line with the **hostid** limited to 12345678. Specification can be one of the following: **VERSION, HOSTID, EXPDATE, KEY, VENDOR_STRING, ISSUER**

If no feature is specified, **lmdiag** will operate on all features in the license file(s) in the path. **lmdiag** will first print information about the license then attempt to check out each license. If the checkout succeeds, **lmdiag** will indicate this. If the checkout fails, **lmdiag** explains the reason for the failure. If the checkout fails because **lmdiag** cannot connect to the license server the 'extended connection diagnostics'.

Extended connection diagnostics attempt to connect to each port on the license server node and can detect if the port number in the license file is incorrect. **lmdiag** will indicate each port number that is listening and indicate whether it is an **lmgrd** process. If **lmdiag** finds the vendor daemon for the feature being tested it will indicate the correct port number for the license file to correct the problem.

A4 LMDOWN

lmdown allows a graceful shutdown of all license daemons (both **lmgrd** and all vendor daemons) on all nodes.

The syntax is:

lmdown [-c license-list] [-vendor name] [-q] [-all] [-force]

- c license-list** path to the file(s) to shutdown
- vendor name** If -vendor name is used, only this vendor daemon will be shutdown, and **lmgrd** will not be shutdown.
- q** does not prompt 'Are you sure?'
- all** If multiple servers are specified, automatically shuts down all of them. -q is implied with -all.
- force** If licenses are borrowed, **lmdown** runs only from the machine where the license server system is running, and then only if the user adds -force.

If **lmdown** encounters more than one server (for example if -c specifies a directory with many *.lic files) and -all is not specified, a choice of license server systems to shut down is presented.

When shutting down a three-server redundant license server system, there is a one-minute delay before the servers shut down. **lmdown** shuts down all three license server systems of a set of redundant license server systems. If you need to shut down one of a set of redundant license server systems (not recommended because you are left with two points of failure), you must kill both the **lmgrd** and vendor daemon processes on that license server machine.

You can protect the unauthorized execution of **lmdown** when you start up the license server manager, **lmgrd**. Shutting down the servers causes users to lose their licenses.

Note: On UNIX, do not use kill -9 to shut down license server systems. On Windows, if you must use the Task Manager to kill the FLEXnet Licensing service, be sure to end the **lmgrd** process first, then all the vendor daemon processes. On UNIX, do not use kill -9 to shut down license server systems. On Windows, if you must use the Task Manager to kill the FLEXnet Licensing service, be sure to end the **lmgrd** process first, then all the vendor daemon processes.

A5 LMHOSTID

lmhostid is used to print the correct hostid value on any machine supported by FLEXlm.

The syntax is:

lmhostid [-n] [type] [-utf8]

-n Only the hostid, itself, is returned as a string, which is appropriate to use with HOSTID= in the license file. Header text is suppressed.

type the type of hostid to print. Type must be one of:

On Windows, it is the system name or, in the case of a terminal server environment, the terminal server client name. On UNIX, it is in the form /dev/ttyxx or the X-Display name.

Platform Dependent Hostids

- ether** Ethernet address.
- string** String id.
- vsn** Volume serial number. (Windows platforms only)
- flexid** Parallel or USB FLEXid dongle identification. This is applicable only for those platforms that support FLEXid dongles. See "FLEXnet Licensing Hostids" for a complete list.
- long** 32-bit hostid.

Platform Independent Hostids

- user** Current user name.
 - display** Current display name. On Windows, it is the system name or, in the case of a terminal server environment, the terminal server client name. On UNIX, it is in the form /dev/ttyxx or the X-Display name.
 - hostname** Current host name.
 - internet** IP address of current platform in the form ###.###.###.###.
- utf8** The hostid is output as a UTF-8 encoded string rather than an ASCII string. If the hostid contains characters other than ASCII A through Z, a through z or 0 through 9, use this option with lmhostid. To view a correct representation of the resulting hostid use a utility (such as Notepad) that can display UTF-8 encoded strings.

Example output is:

lmhostid - Copyright (c) 1989, 2002 Macrovision Corporation

The FLEXlm hostid of this machine is "69021c89".

A6 LMINSTALL

lminstall is designed primarily for typing in decimal format licenses to generate a readable format license file

lminstall [-i {infile | -}] [-o outfile] [-overfmt {2, 3, 4, 5, 5.1, or 6}] [-odecimal]

Normally users will simply type **lminstall**. The user is first prompted for the name of the output license file. The default name is today's date in yyyyddmm.lic format. The file should be moved to the application's default license file directory if specified. Otherwise, the user can use **LM_LICENSE_FILE** to specify the directory where the *.lic files are located.

Decimal format input is verified by checksum of each line. To finish entering, type Q on a line by itself, or enter 2 blank lines.

If `infile' is a dash ('-'), it takes input from stdin. When '-i' is used, default output is stdout; otherwise if -o is not specified **lminstall** prompts the user for an output file name.

A6.1 lminstall as a conversion tool

lminstall can alternatively be used to convert licenses between decimal and readable format, and between different versions of FLEXlm license formats.

To convert from readable to decimal:

lminstall -i infile -o outfile -odecimal

To convert to FLEXlm Version 2 format:

lminstall -i infile -o outfile -verfmt 2

Conversion errors are reported as necessary. **lminstall** has a limit of 1000 lines of input.

A7 LMNEWLOG

The **lmnewlog** utility switches the report log file by moving the existing report log information to a new file then starting a new report log with the original report log file name.

If report logs are rotated using **lmnewlog** instead of **lmswitchr** the filename does not have to be changed in the REPORTLOG line of the vendor's options file (requires a v7.1+ vendor daemon).

The syntax is:

lmnewlog [-c license_file_list] feature renamed_report_log

or

lmnewlog [-c license_file_list] feature renamed_report_log

-c license_file_list Use the specified license file(s)
feature Any feature in this license file
vendor Vendor daemon in this license file
renamed_report_log New file path where existing report log information is to be moved.

A8 LMPATH

The **lmpath** utility allows direct control over FLEXnet license path settings; it is used to add to, override, or get the current license path settings.

The syntax is:

lmpath {-add | -override} {vendor | all} license_file_list

-add Prepends license_file_list to the current license-file list or creates the license file list, if it doesn't exist, initializing it to license_file_list. Duplicates are discarded.

-override Overrides the existing license-file list with license_file_list. If license_file_list is the null string, "", the specified list is deleted.

lmpath -override all ""

Deletes the value of LM_LICENSE_FILE.

lmpath -override vendor ""

Deletes the value of VENDOR_LICENSE_FILE.

vendor A vendor daemon name; effects the value of VENDOR_LICENSE_FILE.

all Refers to all vendor daemons; effects the value of LM_LICENSE_FILE.

license_file_list A colon-separated list on UNIX and a semi-colon-separated list on Windows. If license_file_list is the null string "" the specified entry is deleted.

Note: lmpath works by setting the FLEXnet Licensing registry entry on Windows or \$HOME/.flexlmrc on UNIX. lmpath works by setting the FLEXnet Licensing registry entry on Windows or \$HOME/.flexlmrc on UNIX.

To display the current license path settings, use:

lmpath -status

The following is displayed:

lmpath - Copyright (C) 1989-2002 Macrovision Corporation
Known Vendors:
demo: /counted.lic:/uncounted.lic
Other Vendors:
/usr/local/flexlm/licenses/license.lic

Note: that where the path is set to a directory, all the *.lic files are listed separately.

A9 LMREMOVE

lmremove allows the system administrator to remove a single user's license for a specified feature. If the application is active, it re-checks out the license shortly after it is freed by **lmremove**.

The syntax is:

lmremove [-c license_file_list] feature user host display

or

lmremove [-c license_file_list] -h feature server_host port handle

-c license_file_list Specify license file(s)

feature	Name of the feature checked out by the user
user	Name of the user whose license is being removed as reported by lmstat -a
host	Name of the host the user is logged into, as reported by lmstat -a
display	Name of the display where user is working, as reported by lmstat-a
server_host	Name of the host on which the license server system is running
port T	CP/IP port number where the license server system is running, as reported by lmstat -a
handle	License handle, as reported by lmstat -a

The user, user_host, display, server_host, port, and handle information must be obtained from the output of **lmstat -a**.

lmremove removes all instances of user on user_host and display from usage of feature. If the optional -c license_file_list is specified, the indicated file(s) is used as the license file.

The `-h` variation uses the `server_host`, port, and license handle, as reported by `lmstat -a`. Consider this example `lmstat -a` output:

```
joe nirvana /dev/tty5 (v1.000) (cloud9/7654 102), start Fri 10/29 18:40
```

In this example, the user is "**joe**" the user host is "**nirvana**" the display is "**/dev/tty5**" the server host is "**cloud9**" the TCP/IP port is "**7654**" and the license handle is "**102**"

To remove this license, issue one of the following commands:

```
lmremove f1 joe nirvana /dev/tty5
```

or

```
lmremove -h f1 cloud9 7654 102
```

When removing by handle, if licenses are grouped as duplicates, all duplicate licenses are also removed. If license lingering is set and **lmremove** is used to reclaim the license, **lmremove** starts, but does not override, the license's linger time.

You can protect the unauthorized execution of **lmremove** when you start up the license server manager, `lmgrd`, because removing a user's license is disruptive

A10 LMREREAD

The `lmreread` utility causes the license server manager to re-read the license file and start any new vendor daemons that have been added. In addition, all currently running vendor daemons are signalled to re-read the license file and end-user options files for changes in feature licensing information or option settings. If report logging is enabled, any report log data still in the vendor daemon's internal data buffer is flushed. **lmreread** recognizes changes to server machine host names but cannot be used to change server TCP/IP port numbers.

If the optional vendor daemon name is specified only the named daemon re-reads the license file and its end-user options file (in this case **lmgrd** does not re-read the license file).

The syntax is:

```
lmreread [-c license_file_list] [-vendor vendor] [-all]
```

- c license_file_list** path to the file(s) to find the `lmgrd` server. If more than one file, use colon separator of UNIX, or semi-colon on PC.
- vendor vendor** If `-vendor` name is used, only this vendor daemon will reread the license file. If the vendor daemon is not running, `lmgrd` will start it.
- all** If more than one `lmgrd` is specified, instructs all `lmgrds` to reread.

To disable **lmreread**, the license administrator can use **lmgrd -x lmreread**.

Note: If the **-c** option is used, the license file specified will be read by **lmreread**, not by **lmgrd**; **lmgrd** re-reads the file it read originally. Also, **lmreread** cannot be used to change server node names or port numbers. Vendor daemons will not re-read their option files as a result of **lmreread**.

A11 LMSTAT

The **lmstat** utility helps you monitor the status of all network licensing activities, including:

- Daemons that are running
- License files
- Users of individual features
- Users of features served by a specific vendor daemon
- BORROW licenses borrowed

lmstat prints information received from the license server system; it does not therefore report on unserved licenses such as uncounted licenses. To report on an uncounted license the license must be added to a served license file and the application must be directed to use the license server system for that license file (via @host, port@host or USE_SERVER). Queued users and licenses shared due to duplicate grouping are also not returned by **lmstat**.

The syntax is:

```
lmstat [-a] [-c license_file_list] [-f [feature]] [-i [feature]] [-s[server]
-S [vendor]] [-t timeout_value]
```

- a** Displays everything
- c license_file_list** Uses license_file(s); if more than one file, use colon separator of UNIX, or semi-colon on PC.
- f [feature]** Displays users of feature. If feature is not specified, usage information for all features is displayed.
- i [feature]** Displays information from the FEATURE/INCREMENT line for the specified feature, or all features if feature is not specified.
- s [server_name]** Displays status of all license files listed in \$VENDOR_LICENSE_FILE or \$LM_LICENSE_FILE on server, or on all servers if server is not specified.
- S [vendor]** Lists all users of vendor's features.
- t timeout_value** Sets connection timeout to timeout_value. This limits the amount of time lmstat spends attempting to connect to server.

Note: **lmstat -a** is a potentially expensive command. With many active users, this can generate a lot of network activity, and therefore should not be used too often. **lmremove** requires the output of '**lmstat -a**.'

A12 LMSWITCH

The **lmswitch** utility switches the debug log file written by a particular vendor daemon. It does this by closing the existing debug log for that vendor daemon and starting a new debug log for that vendor daemon with a new file name. It also starts a new debug log file written by that vendor daemon if one does not already exist.

The syntax is:

lmswitch [-c license_file_list] vendor new_debug_log

-c license_file_list Use the specified license file(s)

vendor Vendor daemon in this license file

new_debug_log Path to new debug log file

By default, debug log output from **lmgrd** and all vendor daemons started by that **lmgrd** get written into the same debug file. **lmswitch** allows companies to keep separate log files for different vendors and control the size of their debug log file.

If debug log output is not already directed to a separate file for this vendor daemon **lmswitch** tells the vendor daemon to start writing its debug log output to a file “**new_debug_log**”. If this vendor daemon is already writing to its own debug log **lmswitch** tells the vendor daemon to close its current debug log file and start writing its debug log output to “**new_debug_log**”.

A13 LMSWITCHR

The **lmswitchr** utility switches the report log file by closing the existing report log and starting a new report log with a new file name. It also starts a new report log file if one does not already exist.

The syntax is:

lmswitchr [-c license_file_list] feature new_report_log

-c license_file_list Use the specified license file(s).

feature Any feature in this license file.

new_report_log Path to new report log file

If report logging is not enabled for the vendor daemon **lmswitchr** tells it to start writing its report log output to “**new_report_log**”. If report logging is already enabled for the vendor daemon **lmswitchr** tells the vendor daemon to close its report log file and start writing its new report log output to “**new_report_log**”.

A14 LMVER

lmver reports the FLEXlm version of a library or binary.

The syntax is:

lmver filename

where filename is one of the following:

- the name of an executable file built with FLEXnet Licensing
- lmgrd
- a license administration tool
- a vendor daemon