

Real Time Developer Studio

Installation manual



Contents

Introduction - - - - -	2
Description	2
FLEXIm architecture	2
PragmaDev License Manager	4
C debugger	5
Windows installation - - - - -	6
Real Time Developer Studio	6
FLEXIm	6
FLEXIm server	6
FLEXIm client	7
PLM	9
PLM server	9
PLM client	10
PLM management	10
MinGW	12
Tornado	12
UNIX installation - - - - -	13
Real Time Developer Studio	13
RTDS fonts	13
FLEXIm	13
FLEXIm server	13
Production install	13
Manual launch	14
FLEXIm client	14
PLM	14
PLM server	14
PLM client	15
PLM management	15
Tornado	17
GNU environment	17
Installation	17
Installation on Solaris	18
Installation on Linux	18
Configuration	18

1 - Introduction

1.1 - Description

Real Time Developer Studio installation includes the installation of the following items:

- *Real Time Developer Studio*
- FLEXlm license manager
- PragmaDev license manager (PLM)
- An optional C or C++ compiler with its associated debugger supported by *Real Time Developer Studio* such as Tornado, gdb, Green Hills Multi, or Tasking Cross View Pro. Please note MinGW compiler and debugger are included in the Windows installation.

Real Time Developer Studio can either use FLEXlm licenses or PLM licenses. When requesting your license you may ask which license manager you want to use. Installation of only one of the license manager is necessary.

The following paragraphs describe both the FLEXlm and PLM architecture and installation procedures. It also covers the installation of the GNU environment and some configuration aspects of Tornado. The installation process is platform dependent and is described in section 2 - "Windows installation" on page 6 and in section 3 - "UNIX installation" on page 13.

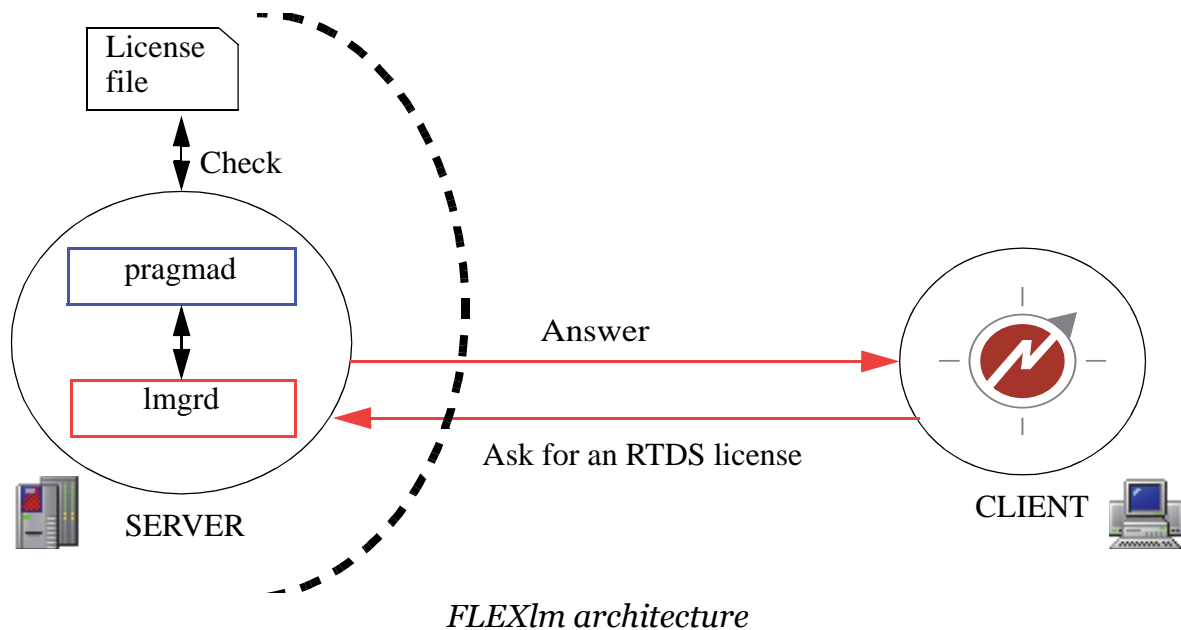
1.2 - FLEXlm architecture

Real Time Developer Studio uses the FLEXlm licence manager version 11.5. FLEXlm license manager is based on a client-server architecture. The client is embedded in *Real Time Developer Studio* while the server must be started separately before the client. The server is the combination of two executable files:

- lmgrd: a generic license server daemon,
- pragmad: PragmaDev specific daemon.

Please note the server and the client can be installed on the same machine.

The number of licenses for RTDS is defined in the license file sent to you by PragmaDev. This file is read by the license server, and each time *Real Time Developer Studio* is started, a license token is asked to the license server. When the maximum number of licenses is reached, the token is refused by the server.



The license file describes the server characteristics and the available licenses:

- the SERVER line specifies the name and the hostid of the server:
SERVER hostname hostid [port number]
- the VENDOR line specifies the name of the PragmaDev daemon:
VENDOR pragmad
- the FEATURE line specifies the maximum number of simultaneous RTDS licenses you can use as well as the version:
FEATURE rtds pragmad version nbOfLicenses SIGN=licenseKey
where *nbOfLicense* indicates the maximum number of licenses available for *Real Time Developer Studio* and *licenseKey* the unique id securing the licenses.
- one or several INCREMENT lines may be added if new licenses are purchased afterwards.

Please note only the hostname and the port number can be modified in the license file. Any other modification might result in failure.

The license file which is received by e-mail, is dedicated to the computer running as a license server. If a FLEXlm license server is already installed, it is recommended to complete the existing licence file. To do so, add the VENDOR and FEATURE lines to the existing file and execute a reread with `lmtools` or `lmutil` utility.

The necessary files to install the FLEXlm license server are located in:

`$RTDS_HOME/share/flexlm/bin/<platform>`

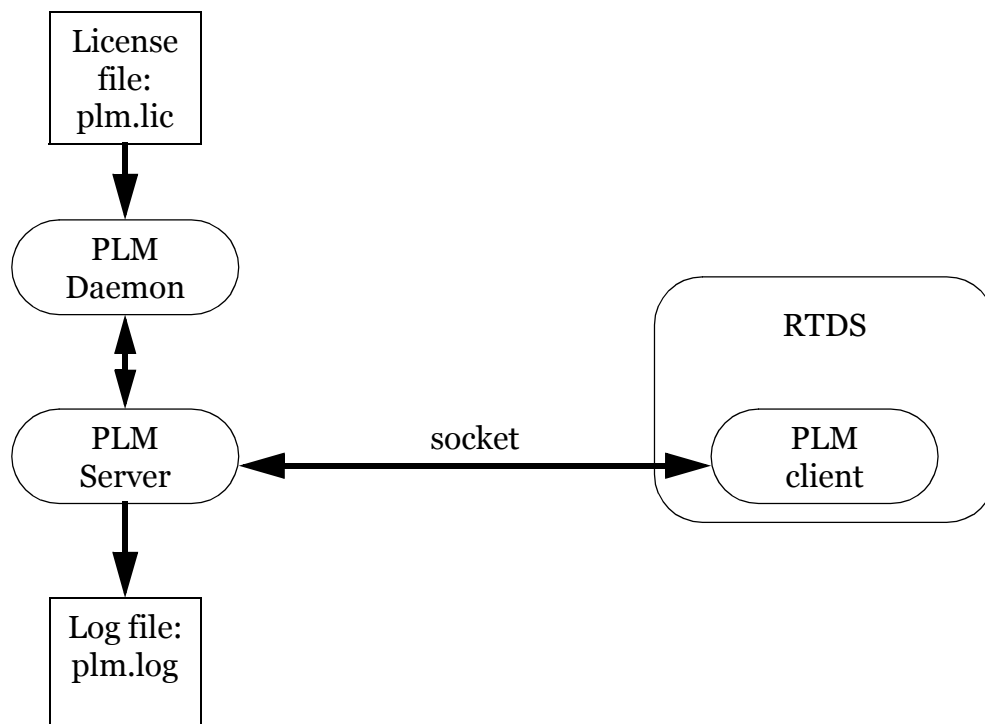
where `$RTDS_HOME` is the directory where you installed RTDS and `<platform>` is windows, solaris, or linux. The files are:

- `lmgrd`: generic daemon,
- `pragmad`: PragmaDev daemon,
- `lmutil`: configuration and administration tool,
- `lmtools`: for Windows only, to setup the license server and install it as a Windows service.

More information can be found in the FLEXlm end user manual located in the \$RTDS_HOME/share/flexlm/doc directory of your installation of *Real Time Developer Studio*.

1.3 - PragmaDev License Manager

The architecture of PLM is similar to the one of FLEXlm. There is a license server that reads the license file and starts the vendor daemons. The vendor daemons will verify the keys in the license file are valid. RTDS embeds a PLM client that connects to the PLM server. The PLM server will grant or not a license to the client.



The PLM Server, PLM daemon, PLM license file, and PLM log file must be in the same directory. The names of the files are:

PLM element	File name
PLM Server	sPlmServer
PLM daemon	pragmadev
PLM license file	plm.lic
PLM log file	plm.log

Table 1: PLM files

The PLM Server and the PLM client can be executed on the same computer. Communications between the different modules is done through sockets. The socket port number to use is defined by the environment variable: PLM_PORT_NUMBER that should be

defined both on the server and on the client side. On the client side the server hostname or IP address should be defined in PLM_SERVER environment variable.

1.4 - C debugger

A C debugger is not required to be able to run *Real Time Developer Studio*: all project management, diagram edition, SDL Z.100 simulator and C code generation features will work without it. However, *Real Time Developer Studio* relies on a C debugger for graphical debugging of the generated code. To use this feature, one of the supported C debuggers must be installed: Tornado, Tasking, Multi, MinGW, or gdb.

MinGW compiler and debugger are included in the Windows installation and are installed with RTDS. The GNU compiler and debugger installation for Linux and Solaris is described as a compatible version is included in *Real Time Developer Studio* distribution.

2 - Windows installation

2.1 - Real Time Developer Studio

The installation is made by running `setup.exe` that should start automatically when the CD-ROM is inserted; if not, the installation will run by double-clicking on the executable which is located in the root directory of the CD-ROM. This setup starts an installation wizard to follow step by step.

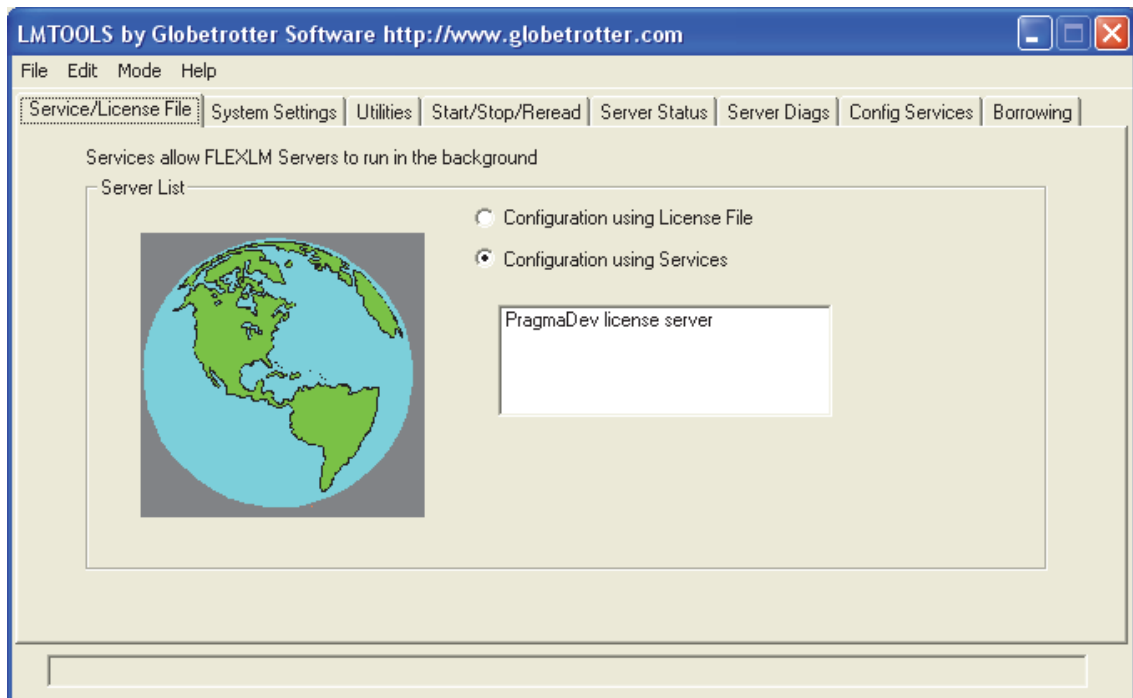
The installation program creates the `RTDS_HOME` environment variable to point to the installation directory and `%RTDS_HOME%/bin` is added to the path.

2.2 - FLEXlm

2.2.1 FLEXlm server

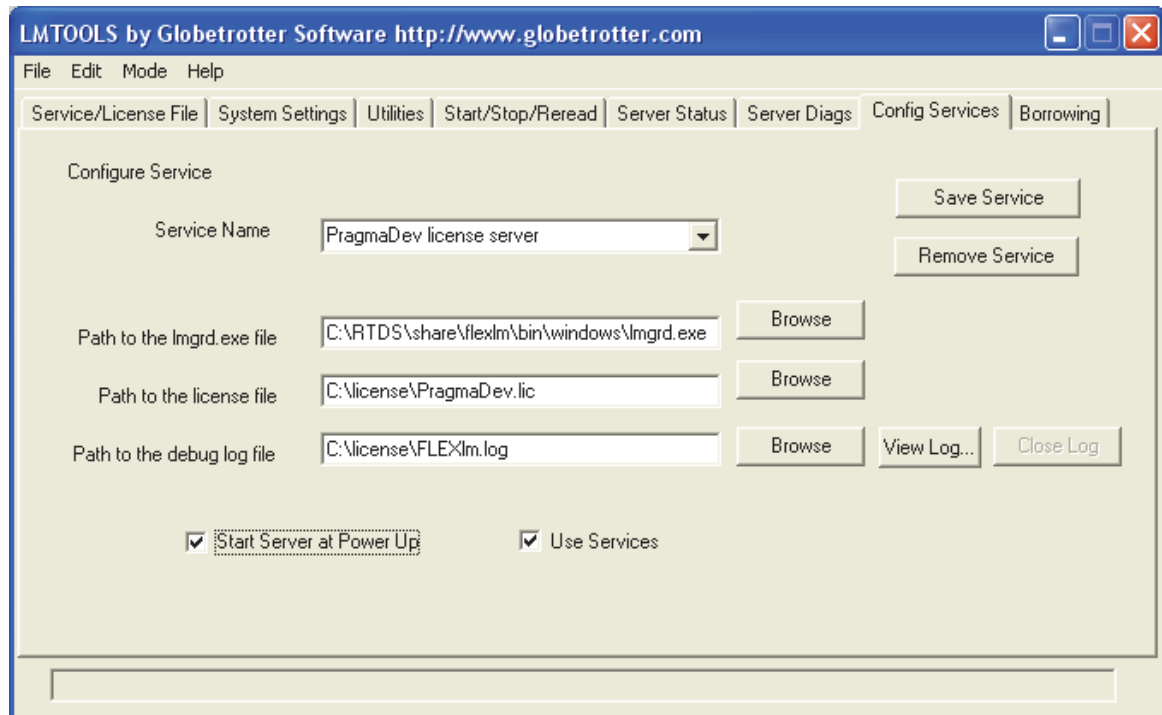
If possible, log in as an administrator to install FLEXlm license server as a Windows service. Otherwise, FLEXlm license server will have to be restarted each time the server is restarted.

Go to `$RTDS_HOME/share/flexlm/bin/windows` directory and start `lmtools`.



- In the "Service/License File" tab check the "Configuration using Services" radio button

- Click the "Config Services" tab



- In the "Service Name" field, type the name of the service that will be defined, for example: PragmaDev license Server
- In the "Path to the lmgrd.exe file" field, enter or browse to the `lmgrd.exe` executable that was installed from the CD-ROM
- In the "Path to the license file" field, enter or browse to the license file containing the license(s) for RTDS
- In the "Path to the debug log file" field, enter or browse to the debug log file. This file will contain all the debug messages from the server.
- If installation is done with administrator privilege, check the Use Services and the Start Server at Power Up boxes for a proper server installation. If installation is done with a non administrator account, the boxes can be left unchecked but the license server will have to be restarted each time the server is powered up.
- Save the new service and go to the "Start/Stop/Restart" tab.
- Click on the "Start Server" button.

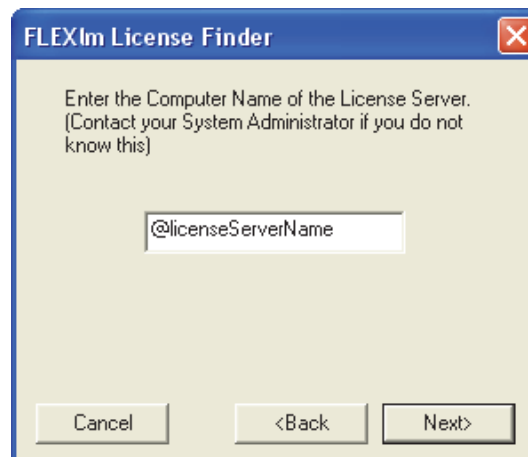
2.2.2 FLEXlm client

The FLEXlm client is embedded in *Real Time Developer Studio* application, so it is started when the application is launched.

The first time *Real Time Developer Studio* is started, a pop up window will ask whether to refer to a license file or to a license server :



The client should preferably refer to the license server: @<license server name> where the license server name is the one specified on the SERVER line in the license file :



Click on Next :



If the pop up window does not show up and if *Real Time Developer Studio* does not work, another FLEXlm license server might be configured on the computer. To solve this

problem, the new FLEX lm server must be declared by executing in a DOS console in
`%RTDS_HOME%/share/flexlm/bin/windows :`
`lmutil lmpath -add pragmad serverLocation`

where *serverLocation* is the name of the server machine (e.g. @myhost), or its IP address, or the full path to the license file.

2.3 - PLM

2.3.1 PLM server

All the necessary files are located in `$RTDS_HOME/share/plm/bin/windows` directory.

The `PLM_PORT_NUMBER` environment variable should be defined as a system variable so that it is always defined. Its value should be set to a valid socket port number on the computer. We recommend a default value of 27042.

The license file you will receive from PragmaDev should be put in the directory where all the files are and named `plm.lic`.

Once you have received the license file, PLM server should be installed as a Windows service so that it is started automatically when the computer boots and even if no user is logged on. The utility `PlmService` will install a service that will start the executables defined in `PlmService.ini`. Make sure the path to `sPlmServer.exe` is correct within that file.

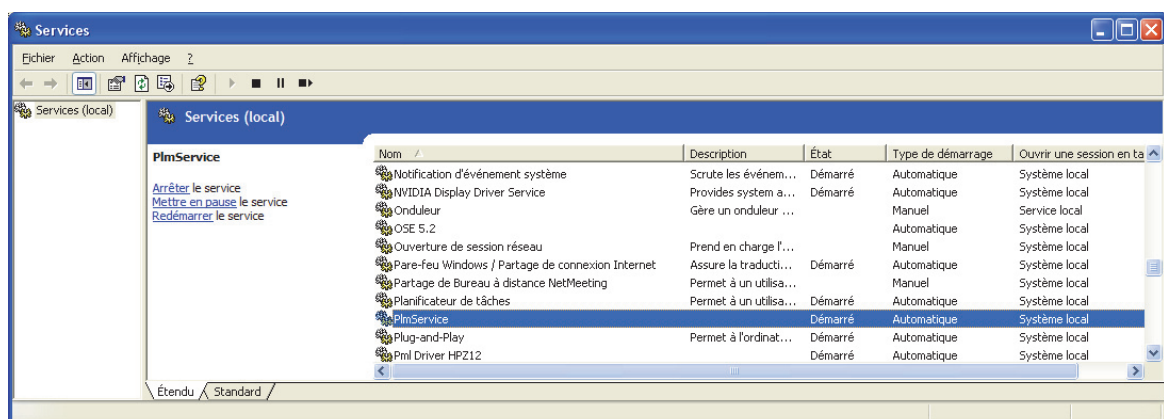
To install the service, type in:

```
PlmService -i
```

To remove the service, type in:

```
PlmService -u
```

Once the service is installed, it will appear in the list of Windows services that can be accessed with the Services utility accessible through the control panel, and the administration tool folder.



The service should be set to start automatically. As installing the service does not actually start it the first time, it is necessary to start it manually in order to check everything is fine: double click on the service and click on the start button.

Note it is also possible to manually start the server, to do so just type: sPlmServer in a DOS console. In that configuration all the log information will also be printed in the console.

2.3.2 PLM client

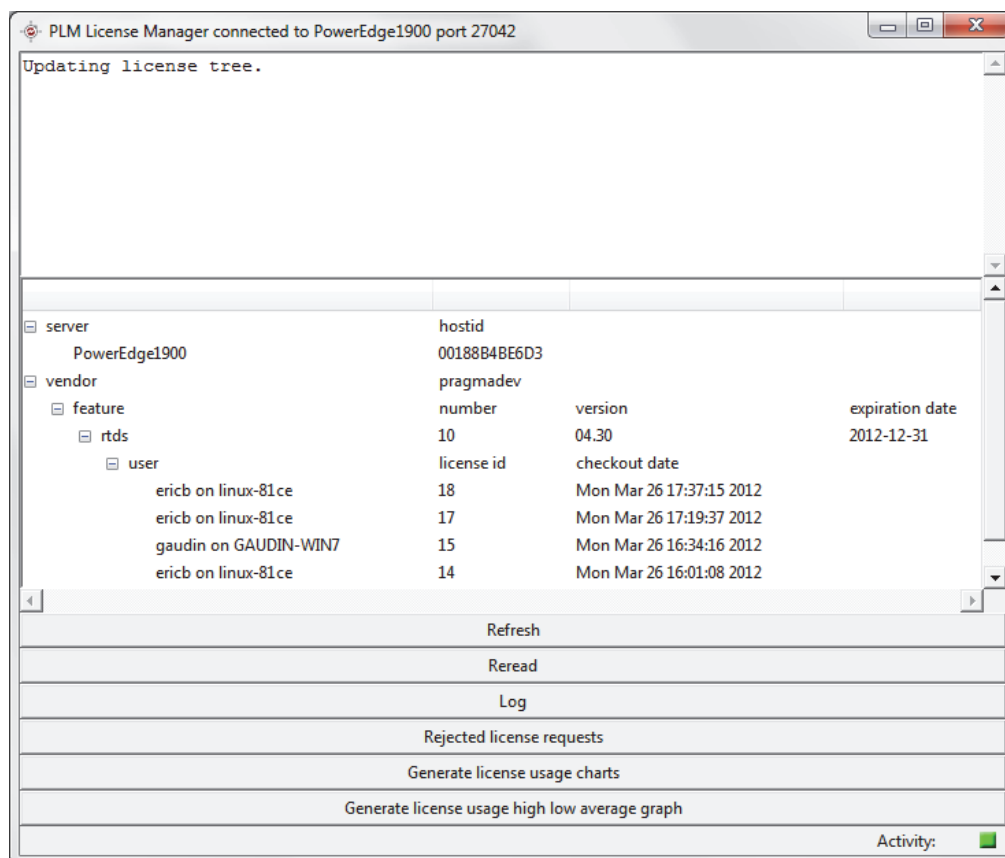
On the client side the following environment variables should be defined:

- PLM_SERVER set to the server hostname or IP address,
- PLM_PORT_NUMBER set to the port number defined on the server.

2.3.3 PLM management

PLM includes a management tool to analyze license usage. It can be run on any computer on the network.

- Set PLM_SERVER to the server hostname or IP address,
 - Set PLM_PORT_NUMBER to the port number defined on the server,
 - run %RTDS_HOME%\share\plm\plm_manager\bin\linux\PLM_Manager.exe
- The following window will open and connects to the license server:

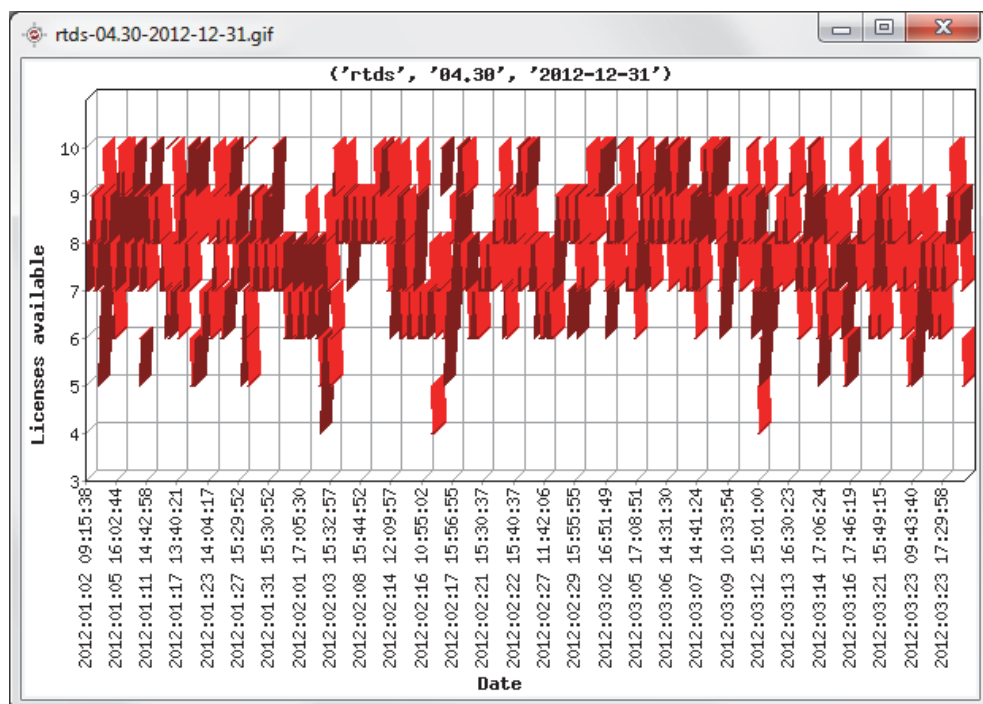


The following features are available:

- Refresh
This will ask the license server some updated information. This is usually not needed since the PLM Manager is refreshed automatically each time the a license is taken or given back.
- Reread

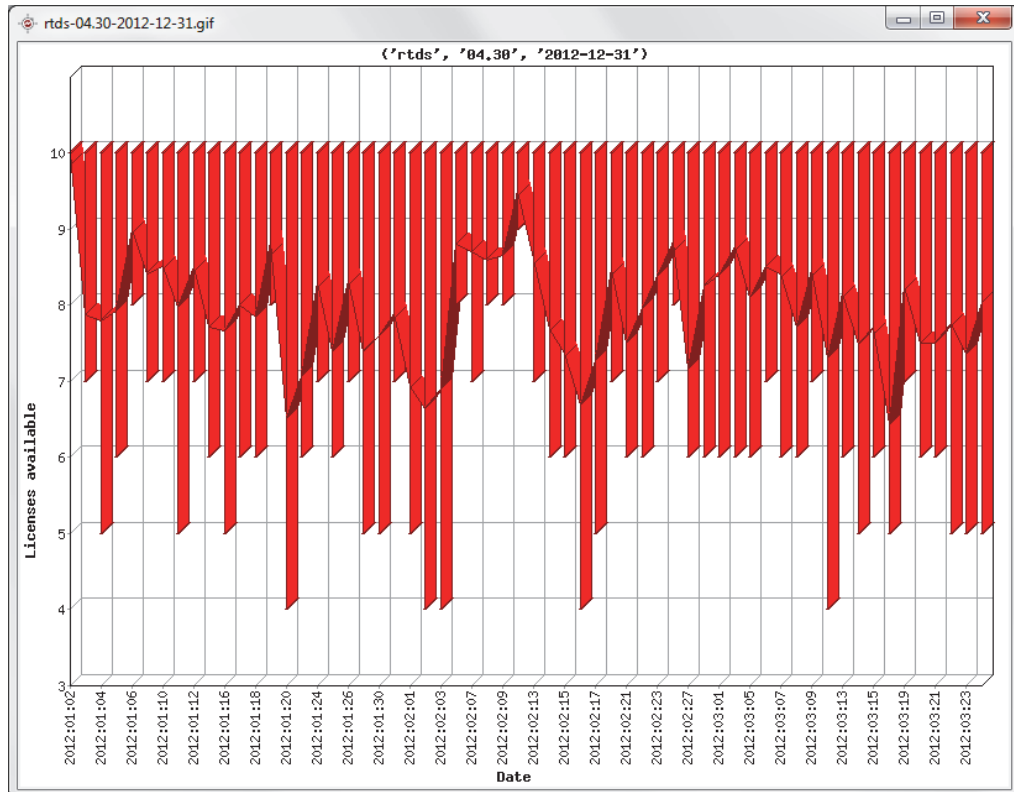
When updating the license file on the server, Reread will read again the license file. When reading a new license file, licenses are not lost if available in the new license file.

- Log
Retrieves the whole log file from the server and displays it on the top of the window.
- Rejected license requests
Retrieves the whole log file - so it might take some time- and displays all license requests that I have rejected.
- Generates license usage charts
Retrieves the whole log file - so it might take some time- and generates a basic license usage chart for each feature based on the log file such as the following one:



Please note the window title is the name of the generated gif file for the graphic.

- Generate license usage high low average graph
Retrieves the whole log file - so it might take some time- and generates a license usage chart for each feature based on the log file with the highs and lows for each day such as the following one:



Please note the window title is the name of the generated gif file for the graphic.

2.4 - MinGW

MinGW 5.1.3 is included in RTDS distribution and is installed with RTDS in `%RTDS_HOME%/share/3rdparty/MinGW` directory and `%RTDS_HOME%/share/3rdparty/MinGW/bin` is added to the path.

C code generation and graphical debugging on the host is therefore possible on Windows straight out of the box.

2.5 - Tornado

In order to use the Tornado integration with *Real Time Developer Studio*, make sure `WIND_BASE` and `WIND_HOST_TYPE` (set to `x86-win32` for Windows) environment variables are defined. If not, Tornado might not be properly installed on your machine. Please refer to WindRiver Tornado installation manual to solve this problem.

3 - UNIX installation

3.1 - Real Time Developer Studio

The installation is made by running the shell script `install.sh` located in the root directory of the CD-ROM. Once the questions asked by the script are answered, the installation will just copy the required files at the specified location. Make sure you have the proper permissions to write to the installation directory.

The installation script installs RTDS for the current platform. If you need to install for another platform (e.g. shared installation on a Unix server for Windows clients), you can give the target platform name as a parameter to the script. The possible platform names are `linux`, `solaris` and `windows`.

Each RTDS client account should then be configured as follow :

- Set `RTDS_HOME` environment variable to RTDS installation directory.
- Add `$RTDS_HOME/bin` to `PATH` and `LD_LIBRARY_PATH` environment variables.

3.2 - RTDS fonts

This part of the installation is optional but recommended to ensure a WYSIWYG behavior of RTDS when printing and exporting diagrams to images. To do so, RTDS should preferably use the same font on the display and when printing and exporting. So the X server on each client must be configured to add RTDS font directory to its font search path.

This is done via the command :

```
xset +fp $RTDS_HOME/share/print/fonts
```

This command must be run on each account after the definition of the `RTDS_HOME` environment variable and after the X server is launched. The command is usually put in the `.xsession` script (if using `xdm`) or the `.xinitrc` script (if not). This is however highly dependent on the X server and the window manager. Please refer to the platform documentation for more information.

3.3 - FLEX/m

3.3.1 FLEX/m server

A proper `FLEX/m` installation requires the license server to be started automatically at power up. `root` privileges are required for such an installation. Still the `FLEX/m` license server can be started without root privilege as explained in paragraph 3.3.1.2 "Manual launch" on page 14.

3.3.1.1 Production install

The `FLEX/m` tools such as `lmgrd`, `lmttools`, `pragmad`, `lmutil` are located in the `$RTDS_HOME/share/flexlm/bin/[solaris|linux]` directory. The way to start the

license server at power up is beyond the scope of this manual, however a template of a service script file automatically starting or stopping the `lmgrd` daemon on server boot / shutdown is provided in the subdirectory corresponding to the machine type of `$RTDS_HOME/share/flexlm/bin/[solaris|linux]`. The name of the service script file is `flexlmd`.

To define the service, just edit the file to uncomment and modify the first lines defining the script variables. Then install the script wherever the service scripts are located on the machine.

3.3.1.2 Manual launch

The license server can be started manually with the following command in `$RTDS_HOME/share/flexlm/bin/[solaris|linux]` directory:

```
> lmgrd -c licensefile
```

where *licensefile* is the file received by email.

3.3.2 FLEXlm client

The *FLEXlm* client is embedded in *Real Time Developer Studio* application, so it is started when the application is launched.

The configuration is done by setting the `LM_LICENSE_FILE` environment variable to:

- Either "`@server`", where `server` is the name or the IP address of the license server as indicated in the license file;
- Or the full name of the license file if visible from the client machine.

The `LM_LICENSE_FILE` variable acts as a path, so you may add the new value to an existing one if any, separating the value with a colon (`:`).

Once the licenser server is up and running, *Real Time Developer Studio* application can be started with the following command in a terminal:

```
> rtds
```

3.4 - PLM

3.4.1 PLM server

All the necessary files are located in `$RTDS_HOME/share/plm/bin/<platform>` directory.

The `PLM_PORT_NUMBER` environment variable should be defined as a system variable so that it is always defined. Its value should be set to a valid socket port number on the computer. We recommend a default value of `27042`.

The license file you will receive from PragmaDev must be put in the directory where all the files are and named `plm.lic`. The PLM server requires a `/tmp` folder to exist.

Once you have received the license file, PLM server should be installed as a Unix daemon to be started automatically when the computer boots. A sample script called `plmd` is provided to be put in `/etc/init.d`. We suggest to set up a symbolic link from `rc5.d/`

S09plmd to that script so that it is started n°9 at startup level 5. In the example script, all the necessary files have been copied to:

/opt/pragmadev/plm/bin

If a PLM server is started while another one is running, the new server will start and the old one will eventually die by itself.

3.4.2 PLM client

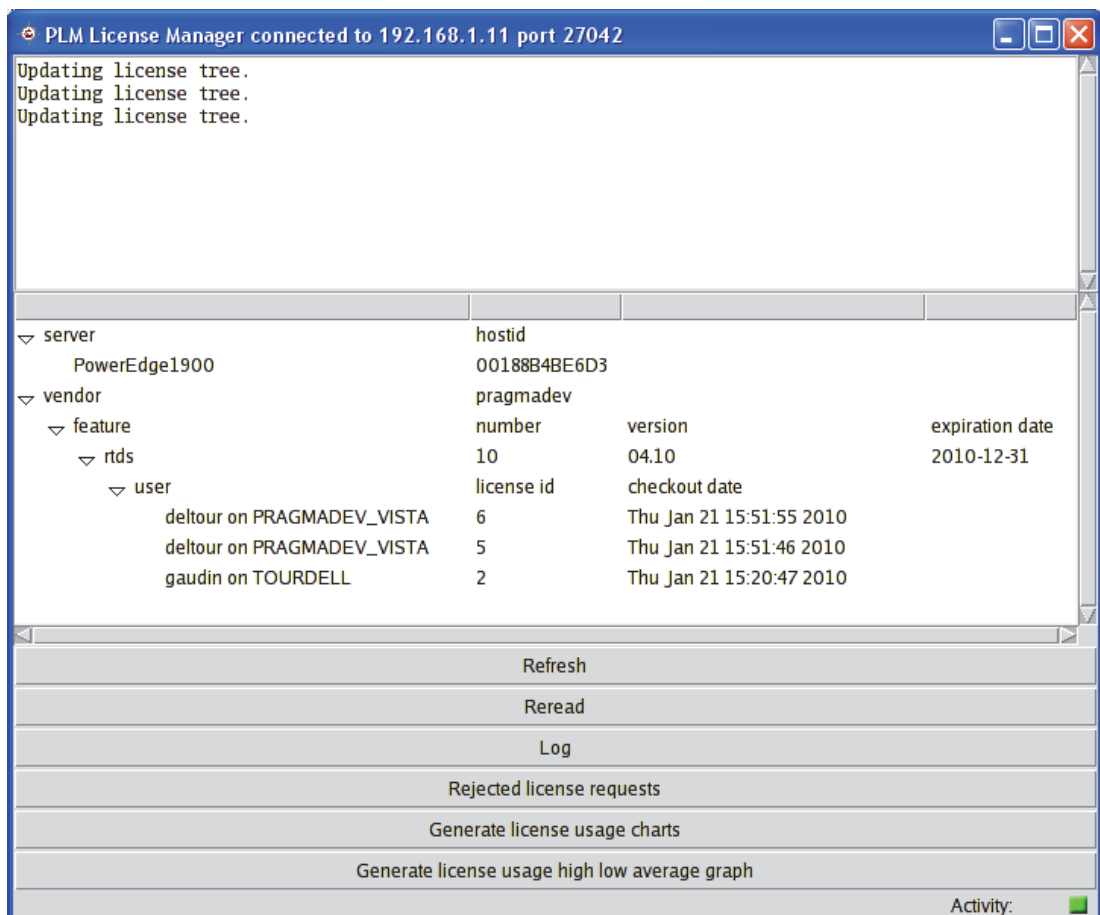
On the client side the following environment variables should be defined:

- PLM_SERVER set to the server hostname or IP address,
- PLM_PORT_NUMBER set to the port number defined on the server.

3.4.3 PLM management

PLM includes a management tool to analyze license usage. It can be run on any computer on the network.

- Set PLM_SERVER to the server hostname or IP address,
 - Set PLM_PORT_NUMBER to the port number defined on the server,
 - run
%RTDS_HOME%\share\plm\plm_manager\bin\win32\PLM_Manager.exe
- The following window will open and connects to the license server:

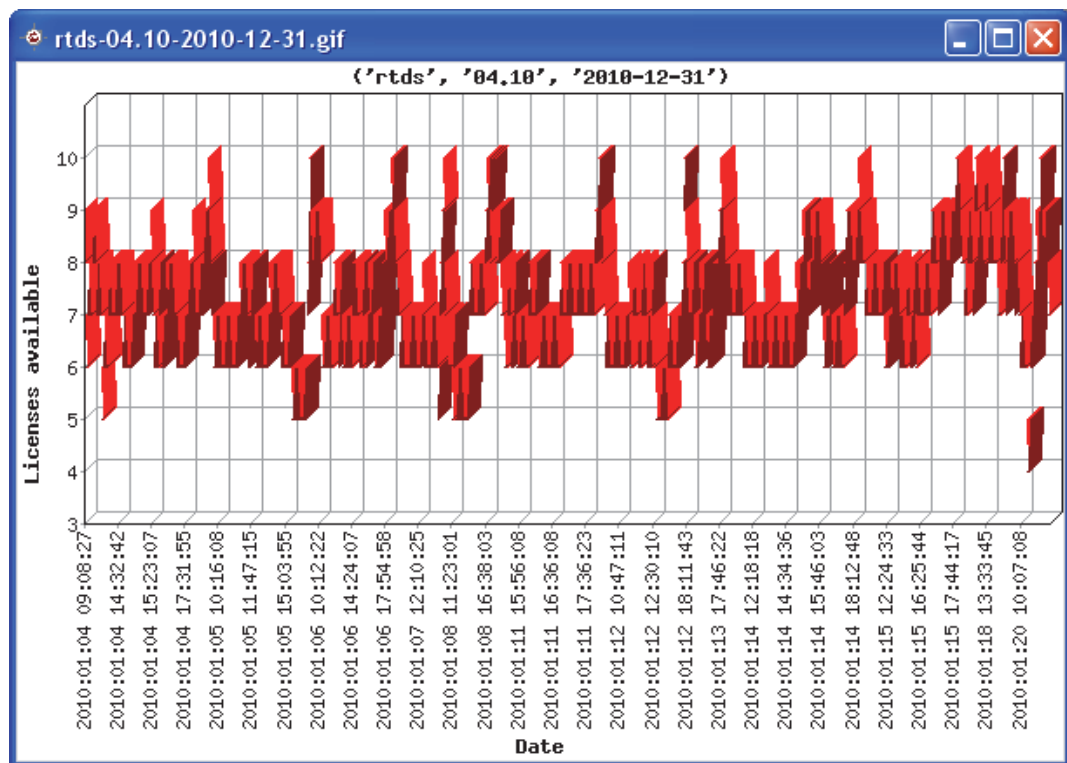


The following features are available:

- Refresh

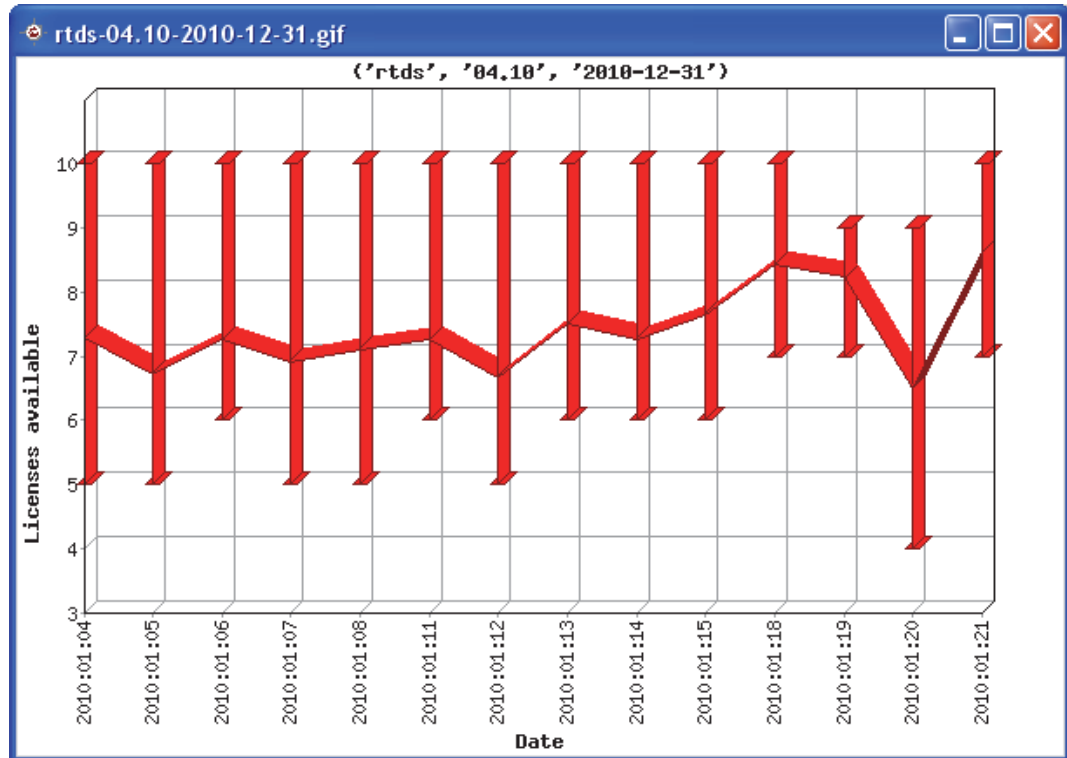
This will ask the license server some updated information. This is usually not needed since the PLM Manager is refreshed automatically each time the a license is taken or given back.

- Reread
When updating the license file on the server, Reread will read again the license file. When reading a new license file, licenses are not lost if available in the new license file.
- Log
Retrieves the whole log file from the server and displays it on the top of the window.
- Rejected license requests
Retrieves the whole log file - so it might take some time- and displays all license requests that I have rejected.
- Generates license usage charts
Retrieves the whole log file - so it might take some time- and generates a basic license usage chart for each feature based on the log file such as the following one:



Please note the window title is the name of the generated gif file for the graphic.

- Generate license usage high low average graph
Retrieves the whole log file - so it might take some time- and generates a license usage chart for each feature based on the log file with the highs and lows for each day such as the following one:



Please note the window title is the name of the generated gif file for the graphic.

3.5 - Tornado

In order to use the Tornado integration with *Real Time Developer Studio*, make sure WIND_BASE and WIND_HOST_TYPE environment variables are defined. If not, Tornado might not be properly installed on your machine. Please refer to WindRiver Tornado installation manual to solve this problem.

3.6 - GNU environment

3.6.1 Installation

In order to use GNU integration with RTDS, gcc, gdb, and make utilities are required. As the behaviour may vary from one version to another, it is recommended to use the one enclosed in RTDS distribution or one of the following versions:

- gcc: 2.95 or higher. Previous versions may work, but have not been tested.
- gdb: 5.2 or higher. As gdb is a critical component of the SDL-RT debugger, it is *strongly* recommended to use this version.
- make: 3.78 or higher.

To retrieve the installed version on your system, type the command name with the `--version` option.

If you need to install any of the 3 utilities, pre-built binaries are available as packages in:

```
$ (RTDS_HOME) /contrib/gnu-<platform>
```

where `$(RTDS_HOME)` is the installation directory and `<platform>` is either `linux` or `solaris`.

The installation process is described in the paragraphs below.

3.6.1.1 Installation on Solaris

`gcc`, `gdb` and `make` are provided as standard Solaris packages. They both install in `/usr/local`.

To install them, become root and execute:

```
pkgadd -d gcc-2.95.3-sol7-sparc-local
```

```
pkgadd -d make-3.78.1-sol7-sparc-local
```

`gdb` is provided as a single executable that should run on any Solaris 7 or 8 platform. It can be copied to any location in the `PATH` (`/usr/local/bin` is a good choice).

More up-to-date version of the packages and `pkgadd` utility may be downloaded from <http://www.sunfreeware.com>.

3.6.1.2 Installation on Linux

The 3 tools are provided as RPM packages. A fourth package containing the C pre-processor, must be installed before `gcc`. They all install in `/usr`.

Installation is done with root privilege:

```
rpm --install cpp-2.96-98.i386.rpm
```

```
rpm --install gcc-2.96-98.i386.rpm
```

```
rpm --install gdb-5.2-2.i386.rpm
```

```
rpm --install make-3.79.1-8.i386.rpm
```

You may have to use `--upgrade` instead of `--install` if a former version is already installed on your machine. More up-to-date versions may be downloaded from <http://rpmfind.net>.

3.6.2 Configuration

If the tools were installed in `/usr`, no additional configuration is needed. Otherwise, make sure that your `PATH` and `LD_LIBRARY_PATH` environment variables include the `bin` and `lib` sub-directories of the installation directory respectively.