



Installation and Licensing Guide

MIRABILIS DESIGN

2010 El Camino Real, #1061

Santa Clara, CA 95050



©2003-2022 Mirabilis Design Inc. All rights reserved.

The information contained herein is subject to change without notice. While every reasonable effort was made to ensure the completeness and correctness of this document, Mirabilis Design Inc. makes no warranty of any kind with regard to this material, including but not limited to any implied warranties. Mirabilis Design Inc. shall not be liable for errors or omissions contained herein or for any damages relating to the use of this material.

VisualSim and VisualSim Architect are registered trademark of Mirabilis Design Inc.

Java and all Java-related titles are trademarks or registered trademarks of Sun Microsystems in the United States and other countries. All other brand or product names may be trademarks of their respective holders.

This document is protected by US and International copyright laws. No part of this document may be reproduced in any manner without prior written consent of Mirabilis Design Inc.

Mirabilis Design Inc.
2010 El Camino Real, #1061
Santa Clara, CA 95050

Revision History

The following table shows the revision history for this document

Date	Version	Revision
Mar 27 2013	13	1
Feb 01 2014	14	1
June 16 2016	16	2
February 17 2017	17	1
February 12, 2018	18	1
January 30, 2019	19	1
April 26, 2019	19	2
October 31, 2019	19	3
March 6, 2020	20	1
August 13, 2021	21	3
January 15, 2022	22	1

Contents

Chapter 1 Requirements	7
Hardware and OS listing	7
Software Requirements	7
Basic Requirements – Java download	7
Web browser	7
Optional Java libraries	7
Interfaces	8
Chapter 2 VisualSim Desktop Installation	10
Getting Host ID	10
Installation Source	10
Installation Sequence	11
VisualSim Desktop Installation Instructions	11
Chapter 3 VisualSim Architect Installation	12
Configuration Setup	13
Chapter 4 Debugging Installation	14
VisualSim License Server Debug	14
VisualSim Architect Install Debug	15
Chapter 5 VisualSim License Manager	17
Installation of Remote License Server	17
Local License Server	18
To update the license file at the License Server	19
License Server as a Windows Service	20
FlexLM License Manager	20
Migration from VisualSim License Server to FlexLM	22
Chapter 6 Mobile License	24
VisualSim Mobile License	24
To obtain a Mobile license using VisualSim License Manager	24
To return a mobile license using the Mobile License Manager	24
To start VisualSim by using a mobile license off the network	25
Debugging Mobile License Issues	25
FlexLM Mobile License	25

Imborrow	25
Initiating Borrowing	25
Determining Borrowed License Status	26
Chapter 7 Starting VisualSim.....	27
Starting VisualSim Architect on Windows	27
Starting VisualSim Architect on UNIX	27
Opening page	27
Pre-Built Models	28
VisualSim Application Models.....	28
Library Models	28
Chapter 8 Verifying the Installation	30
5.1 Open an existing Model	30
Debugging VisualSim Startup	31
Chapter 9 Optional Library Installation.....	32
Emacs Editor Support.....	32
Installing MatLab.....	32
C Interface.....	33
Chapter 10 License Server Statistics	34
Chapter 11 Understanding Errors and Messages	35

Chapter 1 Requirements

Hardware and OS listing

Operating System	Processor Speed	Memory	Disk Space
Windows 10	2.5 GHz Pentium i5 or higher	1512MB min 2048 is best	2000 MB for the application 200 MB user space
Linux	2.5 GHz i5 or Higher	1512MB min 2048 is best	2000 MB for the application 200 MB user space
MAC OS/X	Fastest desktop available	1512MB min 2048 is best	2000 MB for the application 200 MB user space

Software Requirements

Basic Requirements – Java download

Download and install Java Development Kit (JDK). The minimum version must be 17.0.1 with 64-bit. To install Java 17.0.1, go to this Page:

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

Web browser

VisualSim currently supports Safari(Mac), Chrome, Internet Explorer and Firefox only. The documentation and search engine will open in the Web Browser.

Optional Java libraries

If using any of these VisualSim libraries, the Java libraries must be installed.

VisualSim Library	Java Library download
3D Virtual prototyping (or) 3D Interactive creator	Sun/Windows/ Linux HP Apple
Advanced Imaging	Sun/Windows/ Linux Apple
Media framework for Audio and Video	Sun/Linux/Windows

Interfaces

SystemC:

Interface to SystemC is included in the package. For compiling SystemC based VisualSim models you require Visual Studio 2017 on Windows, G++ 4.0 on Linux and the equivalent on Apple Mac.

Verilog:

The user must have licenses of ModelSim Verilog package. These must be purchased separately from VisualSim. To use the Verilog interface, a 64-bit GNU gcc compiler is required. Check here for the [Windows Installer](#), [Linux Installer](#) and [MAC Installer](#). For other Verilog Simulation, please contact Mirabilis Design Inc.

MatLab: The user must provide the MatLab license. The software has been tested with MatLab/Simulink version 2016a and higher.

Application Interface:

VisualSim provides interface to applications in C/C++, One should have machine installed with C/C++ compiler Visual Studio 2017 for Windows and g++ 4.0.x or greater compiler on Unix platforms.

GEM5:

VisualSim GEM5 Wrapper requires the installation of GEM5, which is a separate software. The following steps must be followed for the modification and compiling. GEM5 runs on Linux only. No support is provided for MAC and Windows.

- sudo apt update (optional to update all the packages on Linux)
- mkdir gem5_latest (Create a new directory to save GEM5 install)
- cd gem5_latest (Goto this new directory)
- git clone <https://github.com/gem5-ptolemy/gem5-ptolemy.git> (Download the customized GEM5 version)
- sudo apt install build-essential git m4 scons zlib1g zlib1g-dev libprotobuf-dev protobuf-compiler libprotoc-dev libgoogle-perftools-dev python-dev python
- The following files need to be over written. The modified files are located in VS_AR/doc/GEM5
 - build/drapower/src/Utils.h
 - build/ARM/dev/copy_engine.cc
 - build/ARM/arch/arm/generated/exec-ns.cc.inc

- build/ARM/arch/arm/insts/neon64_mem.hh
- scons build/ARM/gem5.opt -j2

Chapter 2 VisualSim Desktop Installation

Getting Host ID

A host ID is unique to a particular computer and is used by VisualSim to lock licenses to a particular machine. VisualSim license is locked to the physical address (or MAC address) of the computer's network card. If you are using a laptop, ensure that you choose a network device that will not be powered down by your battery management system if it is not in use. The following table lists the operating system commands you can use to determine your host ID for each supported system.

Windows 10

Click Start -> Run and enter cmd. At the Command Window, type `ipconfig /all`. Use the Physical Address for the Wireless Ethernet or the Ethernet interface. The resulting value will look like this 00-1F-3B-CB-46-27.

Linux

At the Terminal Windows, type `"/sbin/ifconfig eth0"` at the command line. Use the colon-separated, hexadecimal number to the right of HWaddr. The resulting value will look like this 8:0:2b:e6:87:59.

Note: With new versions of Linux, `ifconfig` is no longer being shipped. You will need to add the package using the following command-
`sudo apt install net-tools`

Mac OS/X

Open the Apple System Profiler application in `/Applications/Utilities`. Look in the Network overview of the System Profile.

(or) You can also use the command: `netstat -l en0` assuming `en0` is the Ethernet interface. The resulting number will look like this 8.0.2b.e6.87.59.

Installation Source

All software installation instructions are provided online. Visit <https://www.mirabilisdesign.com/getting-started/> to review the documentation and also download software. To download the software, click on the download software link and register. A userid and password will be set to the registered email address.

Installation Sequence

The License Manager, Post Processor and Architect form a single installer. The user can select the products to install. The software must be installed in the following sequence:

License Server

- Install Java SE JDK 17.0.1 or higher
- VisualSim only supports 64-bit OS. So make sure to install the 64-bit version
- Select the License Manager from the VisualSim Architect installer

Desktop

- Install Java SE JDK 17.0.1 or higher
- VisualSim supports 64-bit OS. So, make sure to install the 64-bit version of Java.
- Optional: If installing License Manager for a single user, non-networked license, then install License Manager, VisualSim Architect and VisualSim Post Processor on the same machine
- If a License Server exists, then install only VisualSim Architect and VisualSim Post Processor on each user desktop or laptop.

VisualSim Desktop Installation Instructions

1. Make sure the downloaded file has a .jar extension
3. VisualSim supports both the VisualSim License Manager (all platforms) and Linux FlexLM.
 - If using FlexLM, do not install the License Manager.
 - Only one License Manager is required per License File.

Chapter 3 VisualSim Architect Installation

1. To start the installation process in Graphical mode:
 - a. Double-click the saved file, or,
 - b. Type "java -jar File_Name.jar" at DOS window or Linux/UNIX Terminal.
2. Read and Accept the Readme document and click Next
3. Read and Accept the terms of the License Agreement and click Next
4. Enter the path to install. There should be no space in the path name. When using Windows 7, 8, 10, make sure you create the root directory using Windows Explorer. The security in Windows will not allow the application to automatically create the directory at the root-level.
 - a. If the directory does not exist, a warning message will appear. Accept the message.
5. Enter the Software settings
 - a. Java: Make sure to enter the Java JDK install directory. Do not use the default which is the jre.
 - b. Select the License Manager type
 - i. Default- Uses the VisualSim License Manager
 - ii. FlexLM- Uses the commercial license manager available from Acresto
 - c. License Manager Hostname: Contact your License Administrator for the License Server hostname. If the license server is not the current machine, just leave the default. No suffixes such as mirabilisdesign.com must be entered.
 - i. For FlexLM, enter port number and host name in the format "port@host". Example: 9463@MDI_LS_VS.
6. Select the items to install in the Software Selection Page of the Install.
 - a. If you are only installing the License Manager, make sure you deselect the other two (Architect and Post Processor).
 - b. If installing the Architect and/or the Post processor, deselect the License Manager.
7. Select the icon creation-
 - a. To add items in the Start Menu, select "Create Shortcuts in the Start Menu"
 - b. To add icons on the Desktop, select "Create Additional Short cuts on the Desktop"
 - c. The user must select a group for the Menu items to be listed.
 - d. Note: This feature does not work on the Apple Mac OS.

Configuration Setup

The following information needs to be entered manually after the installation. These variables are not commonly used.

- Path to MatLab directory (variable: MATLAB_PATH)- in the VisualSim.bat (WINDOWS) or VisualSim.sh (UNIX)
- Path to C++ Compiler (variable: VSDIR). The SystemC and CustomC blocks require this update in VSConfig.Properties (Windows) or VSConfig_linux.Properties (Unix) while the C block requires the update in VisualSim.bat and bin/vsconf.bat or bin/vsconf.sh (Unix).
- Path to Custom C, C++, SystemC and Verilog library root. This is the top-level for all user created C++ libraries. (variable: VS_C_Library)- in the VisualSim.bat (WINDOWS) or VisualSim.sh (UNIX)
- Path to the local modeling library directory root. This is the base location for the corporate library management. All sub-models referenced in models are located within this directory. Sub-models or classes are a convenient way to manage reusable components in VisualSim. (variable: VS_Model_Library) in the VisualSim.bat (WINDOWS) or VisualSim.sh (UNIX).
- HTML Export: The parameter HTMLCodeBase in VSConfig.Properties for Windows and VSConfig_Linux.Properties for UNIX is used to setup the base location where the VisualSim Explorer is installed. This would be a URI address and would look like this <http://www.mirabilisdesign.com/new/software>.

Chapter 4 Debugging Installation

Use this table to debug your install:

1. Has the install completed?
 - If not, delete VisualSim install directory and go through the installer again
2. License Server
 - For FlexLM, please contact your FlexLM administrator to ensure that FlexLM is working correctly.
 - For VisualSim License Server, when you start the VisualSim License Server is started, do you see the following on the Window: *“License Manager started successfully. Please DO NOT close this window.”* In the ConsoleMessageLog.txt, you will see the following message: *“License Server started in Normal mode.”* If so, the License Server is working correctly. If not, then there is a error in the installation, go to [License Server Debug Section](#) below.
3. If Step 3 is working correctly but Visual Architect is not coming up correctly, then go to VisualSim Architect Start Debug.

VisualSim License Server Debug

1. Check the files content first
 - View the messages in ConsoleMessageLog.txt and ConsoleDebugLog.txt files in <VisualSim Install Directory>/VS_LM directory. Email these files along with the error message to your technical support.
 - Make sure the license file has been renamed license.lic
 - Check whether you have placed the license file in <VisualSim Install Directory>/VS_LM/com/license directory
 - Check to make sure the hostname in <VisualSim Install Directory>/VS_LM/com/amity/util/Config.Properties file is correct. There are three palces where this same hostname is listed.
 - Open the license file and make sure the “Host ID=08-D4-0C-72-D5-67” matches the Physical Address of this machine.
 - On Linux, make sure the StartServer.sh is executable ie the permissions are 777. If not , run ‘chmod 777 *.sh’ in <VisualSim Install Directory>/VS_AR/.
 - Make sure there are no spaces in the <VisualSim Install Directory> path.
2. System Errors
 - Error 1: “reading license file from path: com/license/license.lic” is the only note in the ConsoleDebugLog.txt file. This indicates that the license file is not in the

correct directory or the file name is (license.lic) not correct. Check both of these issues to resolve this problem. In Windows 8, it is possible for the extension to be repeated, such as license.lic.lic. The second .lic must be removed.

- Error 2: “License Key not Valid” will display in a Dialog Window. This indicates that the license file has been modified. Check the license file that was email to you from Mirabilis Design.
- Error 3: “License Key is Valid. Hostid does not Match” will display in a Dialog Window. This means that the Physical Address in the license file and your machine do not match. This could be because you provided the wrong address or the license file was for a different machine.
- Error 4: “VisualSim Time Server could not be contacted”. This indicates that the Server Name or IP address for the TimeServerHost in the Config.Properties file does not match the machine details. This information would have been provided during the installation process. The file is located in VS_LM/com/amity/util directory.
- Error 5: “License File not Found or Corrupt”. This indicates that the Socket address for the TimeServerPort is being used by another application. Change the value in the Config.Properties in VS_LM/com/amity/util directory. Make sure to communicate this port change information to all the users. They will need to update their files.
- Error 6: If there is no message and the Command Window is blank, then you will see “java.rmi.server.ExportException: Port already in use:” in the ConsoleDebugLog.txt. This indicates that the Socket address for the LicenseServerPort is being used by another application. Change the value in the Config.Properties in VS_LM/com/amity/util directory. Make sure to communicate this port change information to all the users. They will need to update their files.
- Error 7: This is a rare error condition. On the terminal window, you might see “Address already in use”. This indicates that the RMIRegistryPort (1099) is in use by another application. Change the value in the Config.Properties in VS_LM/com/amity/util directory. Make sure to communicate this port change information to all the users. They will need to update their files.

VisualSim Architect Install Debug

1. License Not Available: There are three reasons for this error.
 - The license server has not started. Contact the license server administration to start the license server. If the license is on this machine, then execute <VisualSim Install Directory>/VS_LM/StartServer.bat(.sh).

- There no licenses available. This could indicate that others have consumed the licenses or your prior use license has not been returned to the Server. If it is the second, then try starting the Architect once more.
 - Architect is unable to connect to the License Server. Make sure the <VisualSim Install Directory>/VS_AR/com/amity/util/Config.Properties is the same as the License Server location.
2. Java Exception: This error is not common. The error message is shown in the below figure.

```
C:\Users\randall.richards>"C:\Program Files\Java\jre6\bin\java" -Djava.security.policy=policyAll com.amity.util.StartServer
Exception in thread "main" java.security.AccessControlException: access denied <
java.net.SocketPermission 127.0.0.1:1099 connect,resolve>
    at java.security.AccessControlContext.checkPermission(Unknown Source)
    at java.security.AccessController.checkPermission(Unknown Source)
    at java.lang.SecurityManager.checkPermission(Unknown Source)
    at java.lang.SecurityManager.checkConnect(Unknown Source)
```

- Make sure INSTALL_PATH in CLASSPATH of VisualSim.bat(.sh) file has the right <VisualSim Install Directory path>.

Chapter 5 VisualSim License Manager

VisualSim provides a standard License Server and FlexLM. Flexlm support is limited to Linux. All other platforms and node-locked licenses are supported by the Standard License Server only. Explorer supports the Standard VisualSim License Server. Both licenses enable the user to generate statistics and checkout mobile licenses.

Note: Mirabilis Design provides a script to start the license server as Windows Service.

All VisualSim software users need to first install Java and then install VisualSim. Make sure to specify the jdk path for the Java and the IP address of the license server for the License Manager. On the License Server, the IP address is stored in VS_LM/com/amity/util/Config.Properties. On the client machines, it is located at VS_AR/com/amity/util/Config.Properties.

Installation of Remote License Server

VisualSim License Manager can be installed on a Remote Server to get access on the network. License Manager is included in the Standard installation package and user can opt for only License Manager during installation as Shown in the Figure 3.1.

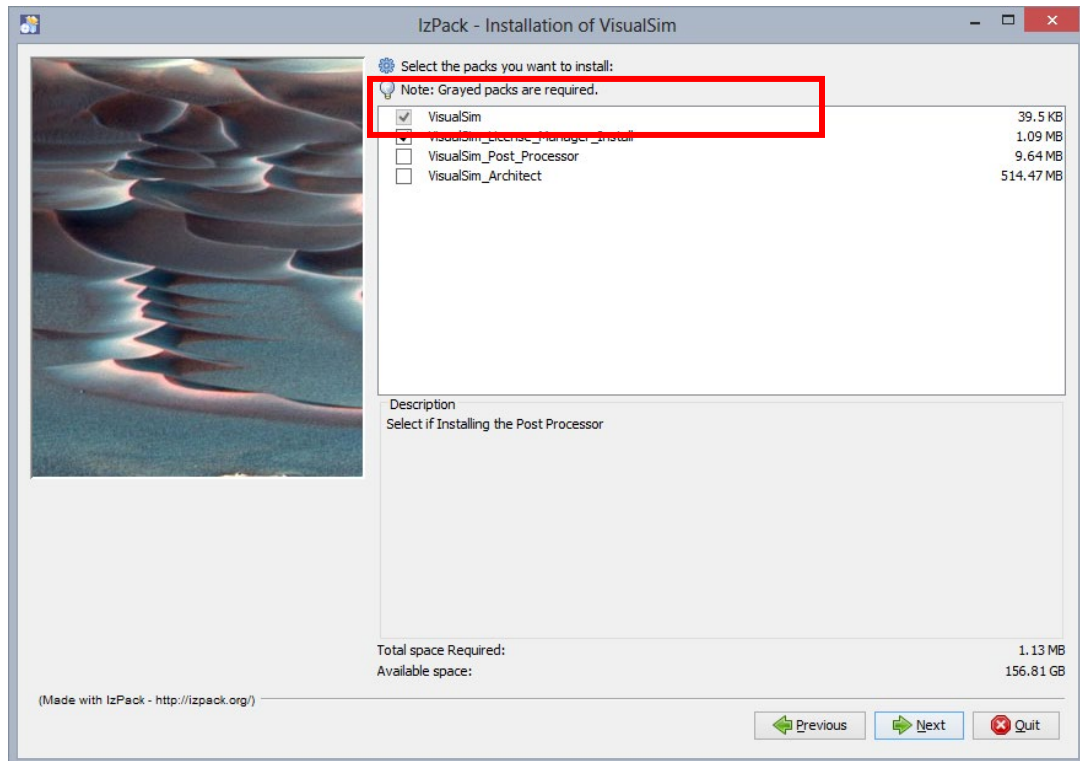


Figure 3.1

After successful installation of License Manager, rename the license file as “license.lic”. The file would have been emailed to you by Mirabilis Design administration. Place this file in VS_LM/com/license directory.

License manager should be running on the Remote Server machine to access license from the Client machine. Edit VS_AR/com/amity/util/CONFIG.PROPERTIES file on the client machine before invoking VisualSim Architect tool. The host name or the Static IP address of the License Manager must be listed in the CONFIG.PROPERTIES. The License Server, Time Server and Mail Server in the License Manager directory, and License Server and the Time Server for the VisualSim Architect/Post Processor must have the same License Manager setting.

Example:

```
TimeServerHost=port_number@server_name  
TimeServerPort=2222  
LicenseServerHost= port_number@server_name  
LicenseServerPort=1900  
RegistryPort = 1099  
MailServerHost= port_number@server_name
```

Local License Server

License Manager can be installed on the local machine alongside with VisualSim Architect. To install License Manager on the local machine one should select License Manager along with VisualSim Architect and PostProcessor during installation as shown in the Figure 3.2.

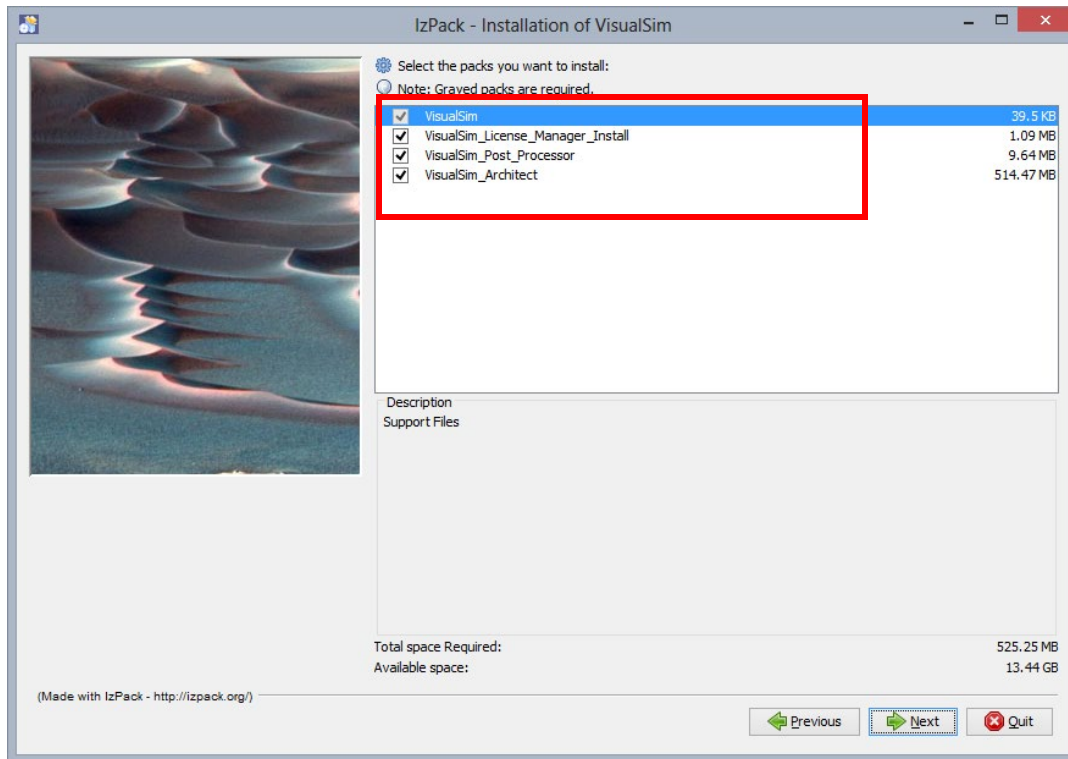


Figure 3.2: License Manager Desktop installation

- License Manager must be running before any application can be started up.
- For VisualSim License Manager
 - Graphical mode
 - Double-click on the License Manager icon on the Desktop for Windows
 - For Console mode
 - Make VisualSim/VS_LM as current directory.
 - Run the command “StartServer.bat” or “StartServer.sh”. This starts a single operation. In the case of UNIX, this operation can be made to run as a background task. In Windows, the DOS Windows must be always open.

To update the license file at the License Server

1. Run ./StopServer.sh
2. Ps aux |grep rmi. Find the process id and do a kill -9
3. Replace the license.lic in VS_LM/com/license directory
4. Run ./StartServer.sh

License Server as a Windows Service

To run the VisualSim License Server as a Windows Service, the following additional steps must be performed.

- Open a Command Window as an Administrator
- cd to <<VisualSim Install Directory>>\VS_LM
- Edit the WinServInstall.bat file. Make sure the first line is pointing to the Java JDK path and not the Java JRE path.
- Type WinServInstall.bat

The Server should be up and running.

- To shutdown the Server, run WinServUnInstall.bat from the <<VisualSim Install Directory>>/VS_LM. Make sure to run it from a Console Window with Administration privileges.

Note: When there is a Java update, the following tasks must be performed by the Administrator.

Copy the following files to (Jre directory)/bin/server/

- Install.bat
- Javaserv.exe
- RunConsole.bat

Remove the existing service

Run install.bat in command window

FlexLM License Manager

FlexLM license can be installed on the Remote Server machine or the local machine during product installation as shown in the Figure 3.3.

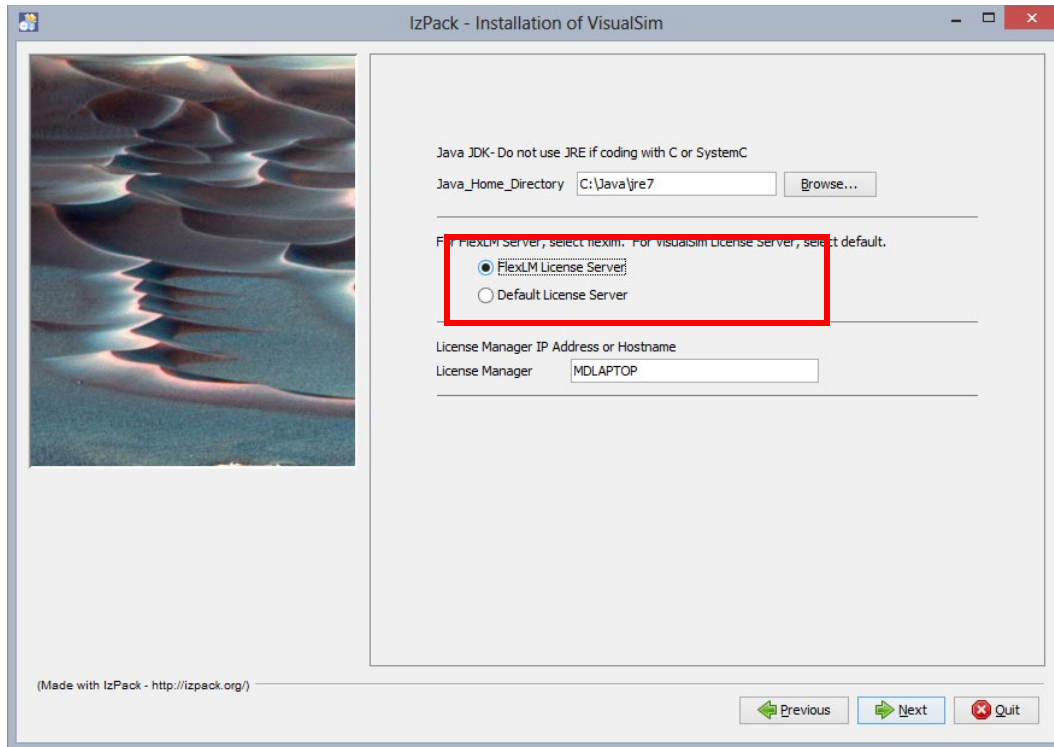


Figure 3.3: FlexLM License Server selection

The following is a simple script showing the start/stop of the FlexLM server with VisualSim:

```

PATH=/usr/bin:/bin
FLEXHOME=/opt/flexlm
LMGRD=$FLEXHOME/lmgrd
LICFILE=$FLEXHOME/nsn_license_temp.lic
LOGFILE=/tmp/flexlm.log
case $1 in
'start')
    rm -f $LOGFILE
    if [ -x $LMGRD ]
    then
        if [ -f $LICFILE ]
        then
            $LMGRD -c $LICFILE -l $LOGFILE
        fi
    fi
    ;;
'stop')
    if [ -x $LMGRD ]
    then
        $LMGRD -x lmdown
    fi
    ;;
*)
    ;;

```

Esac

If the user has Imgrd running and would just like to add the VisualSim licenses, then the administrator can use Imrread. This Imrread utility causes the license daemon to reread the license file and start any new vendor daemons that have been added. In addition all pre-existing vendor daemons will be signaled to reread the license file for any changes in feature licensing information.

Imrread cannot be used to change server node names or port numbers. Its usage is as follows:

```
Imrread [-c license_file] mjalapa
```

where

`-c license_file`

Specifies the license file to operate on. In this case, only Imrread will read the license file specified; Imgrd will reread the license file it originally read at start-up time.

`mjalapa`

Specifies the vendor daemon to signal. If a vendor daemon is specified in this manner, only the named vendor will reread the license file. In this case, Imgrd will not reread the license file (and will not start any new vendor daemons that might have been added).

Migration from VisualSim License Server to FlexLM

As you have already installed VisualSim Architect on your Windows PC/Linux/Mac OS, please make the following changes in the following files

1. VisualSim.bat or VisualSim.sh file under {VisualSim Installation Directory}/VS_AR directory.

Current Setup:

```
::VisualSim FlexLM of Default Mode Execution -Xprof for profiling
java -server -Dvs.lic=default -Duser.language=en -Dvs.lib="%VS_C_Library%" -
Djava.security.policy=bin\policyAll -Xms512m -Xmx1024m -XX:MinHeapFreeRatio=50 -
XX:MaxHeapFreeRatio=50 VisualSim.ModelBuilder.ModelBuilderApplication
::VisualSim startup Execution
```

New Setup

```
::VisualSim FlexLM of Default Mode Execution -Xprof for profiling
java -server -Dvs.lic=flexlm -Duser.language=en -Dvs.lib="%VS_C_Library%" -
Djava.security.policy=bin\policyAll -Xms512m -Xmx1024m -XX:MinHeapFreeRatio=50 -
```

XX:MaxHeapFreeRatio=50 VisualSim.ModelBuilder.ModelBuilderApplication
::VisualSim startup Execution

2. PostProcessor.bat or PostProcessor.sh file under {VisualSim Installation Directory}/VS_AR directory

Current Setup:

"%JAVA_HOME%\bin"\java -Dvs.lic=default -Djava.security.policy=bin\policyAll
VisualSim.plot.plotml.PlotMLApplication

New Setup

"%JAVA_HOME%\bin"\java -Dvs.lic=flexlm -Djava.security.policy=bin\policyAll
VisualSim.plot.plotml.PlotMLApplication

3. flexlmCONFIG.PROPERTIES file located in VS_AR/com/amity/util.

Then, edit flexlmCONFIG.PROPERTIES file located in VS_AR/com/amity/util as follows -
9463@server (Replace "server", with your server name) or 9463@{server IP address}. If port
number 9463 doesn't work, you can use 9465.

4. Now, you can start VisualSim Architect and VisualSim PostProcessor from Desktop Shortcut or
double click on VisualSim.bat and PostProcessor.bat under {Installation Directory}/VS_AR/.

Chapter 6 Mobile License

VisualSim Mobile License

The mobile license provides the client the ability to open the VisualSim Architect graphical environment and use it without being connected to the License Manager. The license is checked out of the server for a period of 1 to 4 days. At the end of the period the license will expire. The user can return the license at any time during this period. During this period, the License manager will have one less license available for checkout.

To obtain a Mobile license using VisualSim License Manager

1. **Pre-requisite:** License server should be running on the network.
2. Using Windows Explorer, go to the VisualSim Architect or VisualSim Post Processor install directory.
3. Run **MobileMgr.bat**.
4. Enable **Application license menu** option.
5. Enter the MAC Address of the client machine.
6. If using Post processor, skip steps 7 and 8.
7. Next Enable **Feature license menu** option. The list of licensed libraries that are available and not expired will be displayed.
8. Select the features that would be used while running VisualSim on Mobile mode. The selected features would be checked out if available. The mobile license file is updated with the checked out feature licenses details.
9. A Mobile application license will be granted and a “**mobile.lic**” file will be created in the current directory.
10. Quit the **Mobile License Manager**.

To return a mobile license using the Mobile License Manager

To check in the mobile application license back to the License Server after returning back to the network, run the **Mobile License Manager** and select the **Checkin Application License** menu option. This checks in the application license and the associated feature

licenses. The mobile license file is removed from the client system. The License Server logs the application usage when the mobile license is checked in.

To start VisualSim by using a mobile license off the network

- **Pre-requisite:** Mobile license should be checked out.
- Start VisualSim Architect and VisualSim Post processor normally. This launches VisualSim in mobile mode. The License server would not be contacted. Mobile license file would henceforth be used for validation.
- The tool usage is logged each time the tool is opened and closed.
- VisualSim is allowed to be used for 24-96 hours of Wall Clock time.

Debugging Mobile License Issues

If you forget to return the license before the expiry, the files may not get deleted on your location. To restart VisualSim after the expiry, delete the following three files:

- MobileLicenseMgrLog
- MobileLicenseUsage
- Mobile.lic

FlexLM Mobile License

Imborrow

If the user has a flexlm license from Mirabilis, use **Imborrow** for the mobile license checkout. The file is located in \$VS/bin directory.

Initiating Borrowing

To initiate borrowing, the user sets the borrow period by running **Imborrow** from the command line. The mobile license is available on Windows Operating System:

```
lmborrow mjalapa enddate [time]
```

For example:

```
lmborrow mjalapa 30-jan-2019
```

or

`lmborrow mjalapa 30-jan-2019 13:00`

lmborrow checks out licenses to the local machine for Architect and all the libraries that were opened on the same day that you run lmborrow. If you run the applications 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.

Argument	Description
<i>enddate</i>	Date the license is to be returned in dd-mm-yyyy format. The checkout lasts until the end of the given end date.
<i>Time</i>	This is an optional field and sets the end time on the enddate.

Table 1: lmborrow Arguments for Initiating Borrowing

Determining Borrowed License Status

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

`lmborrow -status`

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

Chapter 7 Starting VisualSim

Starting VisualSim Architect on Windows

- Click the VisualSim “planet” icon for start server on Window or in the Application Menu.
- Click the VisualSim “planet” icon for VisualSim Architect on Window or in the Application Menu.
- From a Command Window, go to the VisualSim install directory and enter VisualSim.bat.

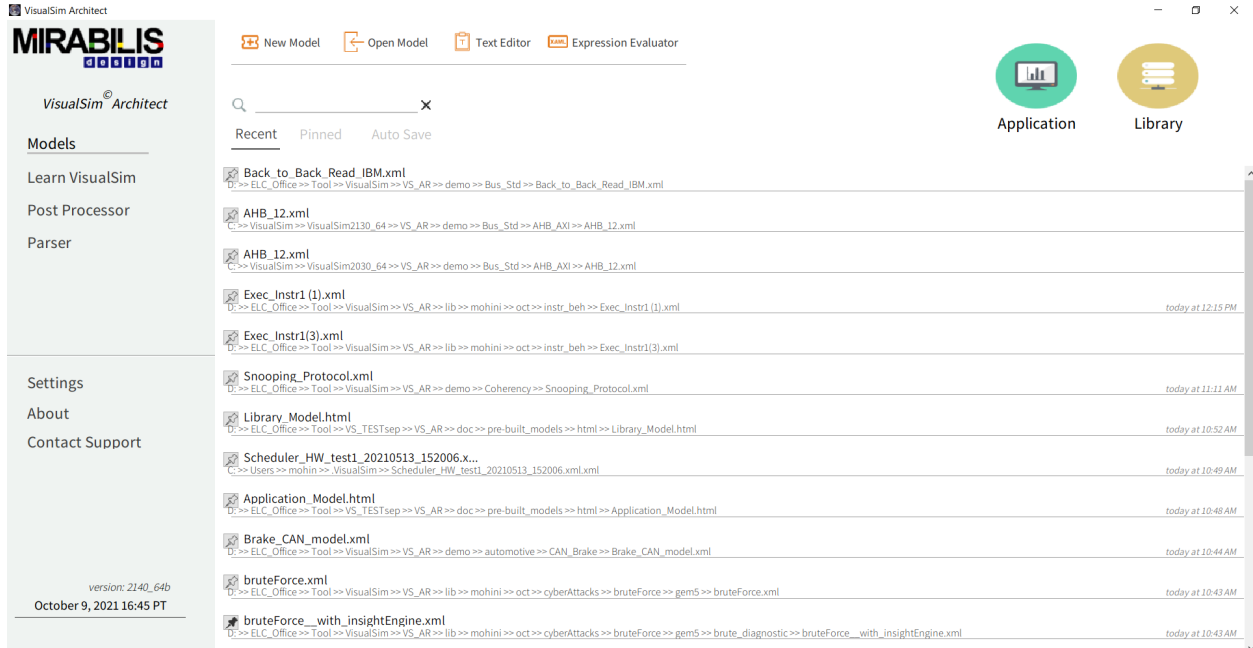
Starting VisualSim Architect on UNIX

Click the Icon that is created on the Desktop, if you selected the Create Shortcut during the Installation.

From a Terminal Window, go to the VisualSim/VS_AR directory and execute ./VisualSim.sh.

Opening page





Pre-Built Models

Under the Pre-Built Models there are two different types of models; Application Demo and Library Demo. Library Demo models show the application of library blocks, the use of RegEx functions and graphical utilities of the Editor. Application Demo models shows the complex system architecture models.

VisualSim Application Models

You can take a tour of common applications using VisualSim. You can go through brief descriptions on the subject of System-Level Exploration and Design, Hardware Architecture Design and Functional and Timing-accurate Software Development. You will also find reference to demo models and documentation explaining the required concept.

Library Models

Basic Graphical Operations

This section focuses on basic operations such as using hierarchical systems, parameters, type system and units. It has several sub-sections as follows:

Performance Modeling

This provides example and demonstration systems on using the various sources, result, resources and behavior, virtual connections, scripting and languages blocks.

Interfaces

In this section you will find demo models for the C, CustomC, SystemC, Verilog, Satellite Toolkit, MatLab, Datagram, Python, Video and audio interfaces.

Using Hardware Architecture blocks

The generic components in the hardware library are explained and used in these examples. These include the processor, DRAM, Cache, I_O, Software Generator, Switch, shared Bus, bridge, switch, storage systems and DMA.

Power Estimation

This section focuses on the use of VisualSim Power Estimation Tools in VisualSim. You will find demo models on Dynamic Voltage Frequency Scaling (DVFS), System and Environment Model and Platform Architecture.

Processor, Bus and Networking Standard Models

In this section you can learn to use PCI, AHB, AXI and Switched Ethernet blocks in VisualSim. You will find demo models on SoC level, Board-Level, Interconnect, NetworkLevel and Processors.

Xilinx FPGA Modeling Toolkit

This section contains models on VisualSim Xilinx FPGA Modeling Toolkit such as Microblaze, Virtex 4-MPMC2, Virtex 4-CoreConnect and Virtex 4-PCI-to-Memory.

Communications Example Systems

In this section you will find demo models on Networking and Wireless Sensor.

Algorithms

This section focuses on signal, image, audio, analog and control systems.

Chapter 8 Verifying the Installation

5.1 Open an existing Model

On the VisualSim Architect File Menu, point to Open File, Select demo → Traffic → Multi_stream_Traffic.xml

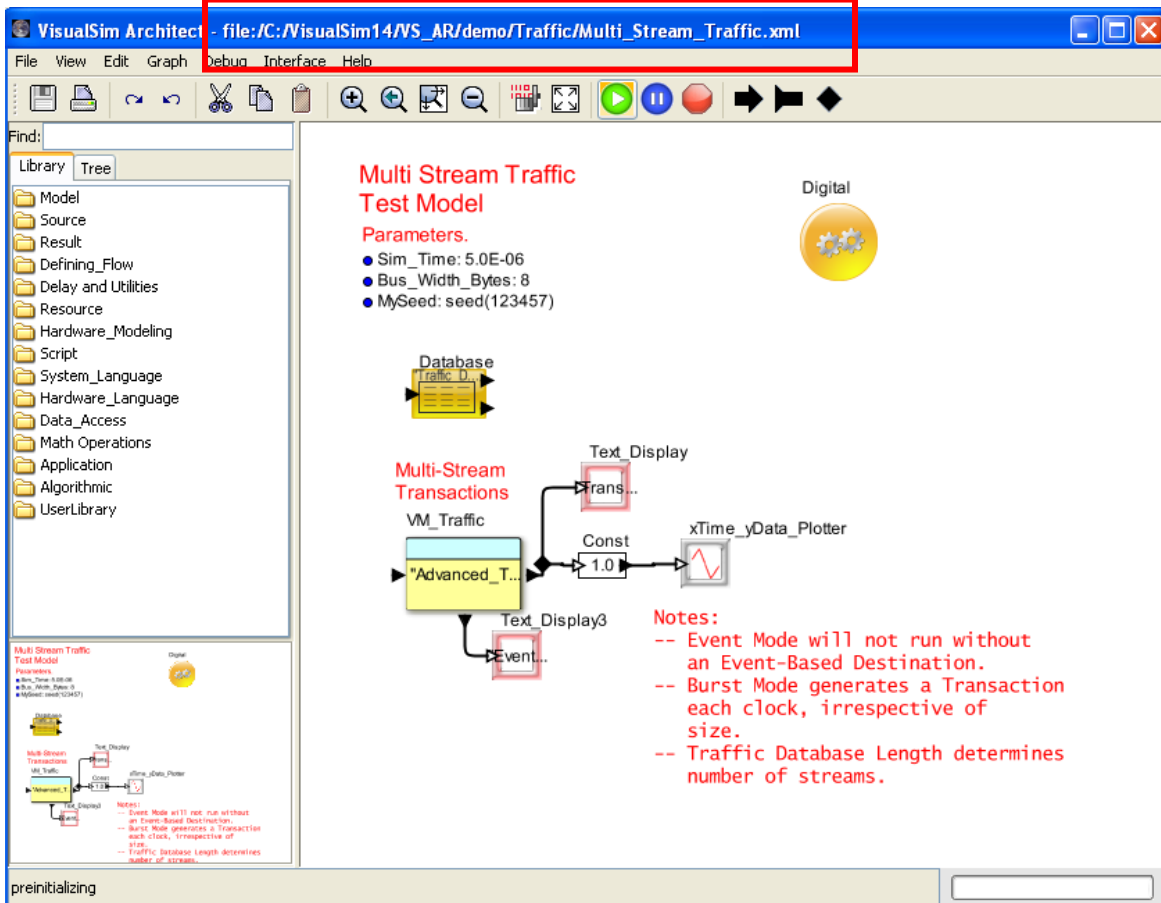



Figure 5.1: VisualSim demo model

5.2 Click on the Run icon  exists below the menu bar. You will find popup graphical viewers that will display results. The status bar at the bottom-left of the tool (VisualSim) will show the current state of the simulation as shown in the figure below:

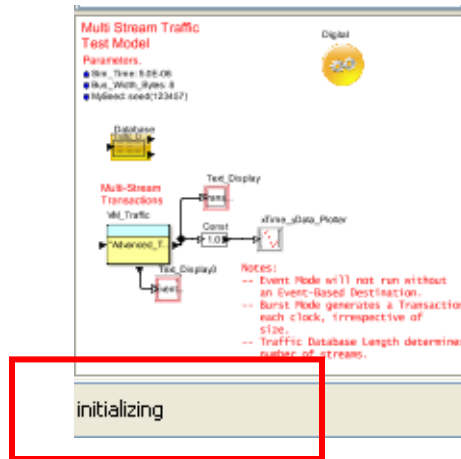


Figure 5.2: Status of Execution

5.3 On successful model execution completion, the status of Execution will be “execution finished”

If the simulation is unsuccessful, then refer the section **“Understanding Errors and Messages”** for debugging.

Debugging VisualSim Startup

If VisualSim Architect does not come up, refer below to identify the possible reasons:

- License Server has not been started
- Run the following lines from the Command Line in order.
 - Set JAVA_HOME = xxx (Windows) or export JAVA_HOME = xxx (UNIX)
 - java -Dvs.lic=default -Djava.security.policy=bin\policyAll
VisualSim.ModelBuilder.ModelBuilderApplication

If this works, then the installation is fine. Contact Mirabilis Design to modify the startup script. The “default” value must be flexlm, if using FlexLM.

Chapter 9 Optional Library Installation

Emacs Editor Support

To use VisualSim with Emacs, two operations have to be completed. The first is to update the VSConfig.Properties or VSConfig_Linux.Properties (Overwrite VSConfig.Properties).

VisualSim.user.texteditor=emacs

To install Emacs, make sure to follow the sequence listed below. Item 8 will ensure that Emacs is installed properly.

1. Go to <http://ftp.xemacs.org/pub/xemacs/binaries/win32/InnoSetup/>
2. Install XEmacs_Setup_21.5.31.exe
3. Install gnuclient_setup.exe
4. Add "C:\Program Files (x86)\XEmacs\XEmacs-21.5.31\i586-pc-win32" to your Path environment variable of VisualSim.bat(or .sh)
5. Create %USERPROFILE%\xemacs directory if it does not exist. %UserProfile% is your Home Directory. There will be a .VisualSim folder in the same directory.
6. Copy file C:\Program Files (x86)\XEmacs\XEmacs-21.5.31\etc\sample.init.el to %USERPROFILE%\xemacs\init.el
7. Append to the init.el file the command as follows: **(gnuserv-start)**
8. Verify the settings - open the init.el file in XEmacs editor from command prompt using the command as follows: **gnuclient -e (find-file user-init-file)(buffer-name)**

Installing MatLab

1. Make sure that you are using VisualSim 64 bit version, Java JDK 7 64 bit. Visual Studio 64 bit
2. Open VisualSim.bat in Edit mode and make sure that you have provided MatLab installation path and Visual Studio Installation path


```
set MATLAB_PATH=C:\Program Files (x86)\MATLAB\R2010a
set VSDIR = C:\Program Files (x86)\Microsoft Visual Studio 12.0\VC
Add %MATLAB_PATH%\bin\win32 to Path variable settings
set Path=%PATH%;%INSTALL_PATH%\VisualSim\matlab; %MATLAB_PATH%;%VERILOGPATH%;
%VERILOGPATH%\win32;%GCCPATH%;%MATLAB_PATH%\bin\win32
```
3. Open VSConfig.properties file and make sure that VSDIR path is updated


```
VSDIR=C:\\Program Files (x86)\\Microsoft Visual Studio 14.0\\VC
```
4. Open VSConfig.properties file and make sure that VSDIR path is updated


```
VSDIR=C:\\Program Files (x86)\\Microsoft Visual Studio 14.0\\VC
```
5. Now Open {VisualSim Install Directory}\\VS_AR\VisualSim\matlab and update Config_Matlab.txt file for your system. This includes setting VS Installation path, Java Path, Visual Studio Path and MatLab path. Save the file as Config_Matlab.bat

6. Open Command Prompt and run Config_Matlab.bat file. Make sure that .lib files are created and no errors are reported.
7. Now Open VisualSim Architect and Open Model
VS_AR/demo/Interfaces/MatLab/MatlabExpression.xml
8. Run the Model. As VisualSim should start MatLab Engine it takes 30 secs to 1 min to start simulation.

Please refer the video for the step by step approach. Link to Video is as follows

<http://www.mirabilisdesign.com/Resources/Videos/VisualSim%20Interface%20with%20MatLab.wmv>

C Interface

1. Open VSConfig.properties file and make sure that AUTOSAR path is correct. Sample Path would look as below;

AUTOSAR=E:\\VisualSim\\VisualSim1620\\VS_AR\\demo\\automotive\\Autosar

2. Open model VS_AR/demo/automotive/Autosar/AUTOSAR_Scheduler/Autosar_Model4a.xml
3. Interfaces --> Generate Wrapper
4. Interfaces --> Compile Wrapper (Make sure that there are no errors)
5. Run the model

Few Other models to test for C Interface are as below

1. demo/Interfaces/CustomC\\Master_Bus_Slave/Master_Bus_Slave.xml
2. demo\\Interfaces\\CustomC\\ApplicationInterfaceTest\\ApplicationInterfaceTest.xml
3. demo\\Interfaces\\CustomC\\ApplicationInterfaceTest4\\ApplicationInterfaceTest4.xml

Chapter 10 License Server Statistics

VisualSim License Server has two files for debugging called ConsoleDebugLog.txt and MessageDebugLog.txt. These provide information on the status, number of active and mobile licenses, any errors during checkout and inactive users. Log_usage.txt provides a summary of each session and a summary every fixed period of time. The time period is defined in the Scheduler.Properties file located in \$VS/com/amity/util directory.

Users can get access to current usage of the FlexLM license server using the lmstat utility. This utility is located in \$VS/bin directory. To generate statistics, first

Open a new DOS Window

Cd \$VS/bin directory

Type "lmstat"

The complete path to the license file is specified.

Now type "lmstat -a -c license_file_path"

This will give you information on the current usage.

Chapter 11 Understanding Errors and Messages

MatLab Errors

1. Unresolved external symbol `_mexPrintf` referenced in `ptmatlab.obj`. This error occurs while compiling matlab interface script.

```

C:\Windows\system32\cmd.exe
Copyright (C) Microsoft Corporation. All rights reserved.

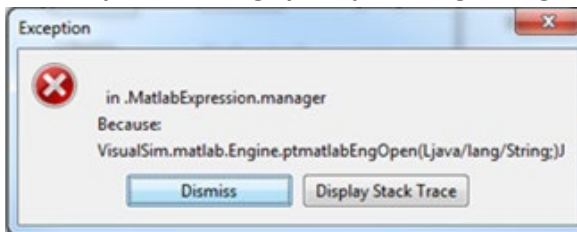
ptmatlab.cc
ptmatlab.cc(707) : warning C4805: '==' : unsafe mix of type 'char' and type 'bool'
in operation
Microsoft (R) Incremental Linker Version 10.00.30319.01
Copyright (C) Microsoft Corporation. All rights reserved.

/out:ptmatlab.dll
/DLL
"/LIBPATH:C:\Program Files\MATLAB\R2010b\extern\lib\win64\microsoft"
libmx.lib
libeng.lib
libmex.lib
ptmatlab.obj
Creating library ptmatlab.lib and object ptmatlab.exp
ptmatlab.obj : error LNK2019: unresolved external symbol _mexPrintf referenced in
function _Java_VisualSim_matlab_Engine_ptmatlabEngOpen@12
ptmatlab.obj : error LNK2019: unresolved external symbol _engOpen referenced in
function _Java_VisualSim_matlab_Engine_ptmatlabEngOpen@12
ptmatlab.obj : error LNK2019: unresolved external symbol _engClose referenced in
function _Java_VisualSim_matlab_Engine_ptmatlabEngClose@24
ptmatlab.obj : error LNK2019: unresolved external symbol _engEvalString referenc
ed in function _Java_VisualSim_matlab_Engine_ptmatlabEngEvalString@20
ptmatlab.obj : error LNK2019: unresolved external symbol _mxGetDimensions_700 re
ferenced in function _Java_VisualSim_matlab_Engine_ptmatlabEngGetArray@20
ptmatlab.obj : error LNK2019: unresolved external symbol _engGetVariable referen
ced in function _Java_VisualSim_matlab_Engine_ptmatlabEngGetArray@20
ptmatlab.obj : error LNK2019: unresolved external symbol _engPutVariable referen
ced in function _Java_VisualSim_matlab_Engine_ptmatlabEngPutArray@28
ptmatlab.obj : error LNK2019: unresolved external symbol _engOutputBuffer referen
ced in function _Java_VisualSim_matlab_Engine_ptmatlabEngOutputBuffer@20
ptmatlab.obj : error LNK2019: unresolved external symbol _mxCreateCellMatrix_700
referenced in function _Java_VisualSim_matlab_Engine_ptmatlabCreateCellMatrix@2
0
ptmatlab.obj : error LNK2019: unresolved external symbol _mxGetData referenced in
function _Java_VisualSim_matlab_Engine_ptmatlabCreateString@24
ptmatlab.obj : error LNK2019: unresolved external symbol _mxCreateCharArray_700
referenced in function _Java_VisualSim_matlab_Engine_ptmatlabCreateString@24
ptmatlab.obj : error LNK2019: unresolved external symbol _mxCreateString referen
ced in function _Java_VisualSim_matlab_Engine_ptmatlabCreateString@24
ptmatlab.obj : error LNK2019: unresolved external symbol _mxGetPr referenced in

```

In this case user was trying to compile MatLab interface with 32 bit version of Visual Studio and 64 bit MatLab version.

2. Error in `ptmatlab.EngOpen(Ljava/lang/String)`



This error occurs when user trying to run VisualSim model having matlab instance without having successful MatLab interface.

Also Check if PATH has `%MATLAB_PATH%\bin\win32` in VisualSim.bat file

3. Can't find MatLab engine.
 - 3.1: On windows platform



This error occurs when multiple instances of MatLab are installed on the same machine and by running `matlab -regserver` on CommandPrompt(Admin) for selected MatLab version will resolve the error,

3.2 On Linux Platform

Error is because of lack of `\bin\csh` in the PATH.

“`System.load(library): VisualSim/matlab/ptmatlab`”

`System.load(libraryPath) FAILED: E:\VisualSim13\VS_AR\VisualSim\matlab\ptmatlab.`

`dll Exception: java.lang.UnsatisfiedLinkError: E:\VisualSim13\VS_AR\VisualSim\matlab\ptmatlab.dll: Can't load IA 32-bit .dll on a AMD 64-bit platform”`

This error occurs when the user compile MatLab interface with one version of Java and Run VisualSim model that is having MatLab module with an another Java version.

4. Command Line Error

“`Ipconfig / all` is not recognized as an internal or external command, operable program or batch file.”

Solution: To clear this error go to My computer->Right Click “Properties”->Advanced Settings tab ->Environmental Settings->System Settings->Add to PATH “`C:\WINDOWS\system 32;`”.

