

# Integration with MATLAB

You can use MATLAB® to evaluate expressions written in MATLAB syntax in Cameo Simulation Toolkit. You must install MATLAB® first and set up your system to call and use it in Cameo Simulation Toolkit.



## Note

- For successful integration, you must use either the 64-bit or 32-bit version of MATLAB (Version 2012a or later) to align it with the 64-bit or 32-bit version of modeling tools, e.g., MagicDraw or Cameo Systems Modeler.



## Warning

MATLAB® 2014a on Mac OS is not capable of evaluating and returning values via Simulation Console, which causes MagicDraw to freeze. To solve this problem, please consider upgrading MATLAB® 2014a to 2014b or later.

To disable OS X El Capitan's System Integrity Protection (SIP)

1. Restart your Mac.
2. As soon as the screen turns black, hold down the **command+R** keys to access the Recovery Partition.
3. Release the keys once the Apple logo appears on your screen.
4. Click the **Utilities** menu and select **Terminal**.
5. Type `csrutil disable` into the Terminal window and press **Return**.
6. The Terminal should display a message that SIP was disabled.
7. Restart your Mac for the changes to take effect.
8. Run MagicDraw/Cameo System Modeler and integrate it with MATLAB®.

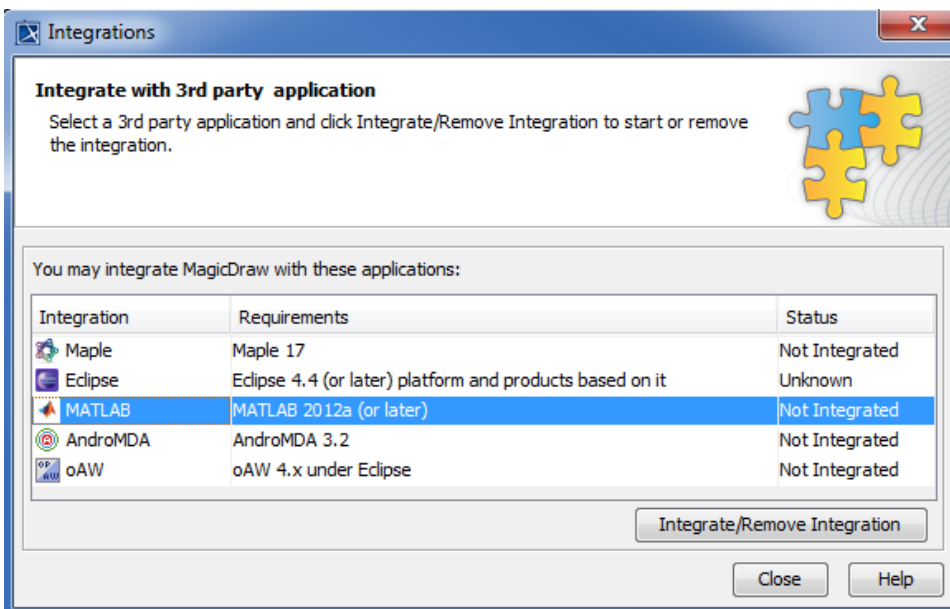


## Note

To enable the **Integrations** menu under the **Tools** main menu, you must change the perspective to **System Engineer** or **Software Architect** by doing the following:

To integrate MagicDraw or Cameo Systems Modeler with MATLAB® (on Microsoft Windows or Mac OS X)

1. From the main menu, select **Options > Perspectives > Perspectives**. The **Select Perspective** dialog opens.
2. From the main menu, click **Tools > Integrations**. The **Integrations** dialog opens.
2. Select **System Engineer** or **Software Architect** and click **Apply**. The **Integrations** menu will be enabled.



2. Select **MATLAB®** and click **Integrate/Remove Integration**. The MATLAB® directory selection dialog opens.
3. Specify the directory where you installed MATLAB® and click **OK**.

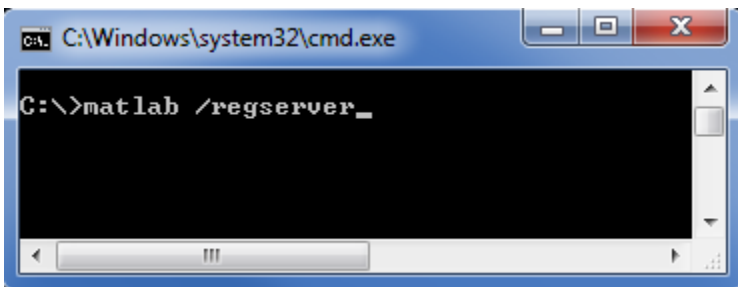


## Note

You can also integrate MagicDraw with Mageia Systems Modeler with MATLAB® manually using the following steps with the administrator's privileges, and then try to integrate again.

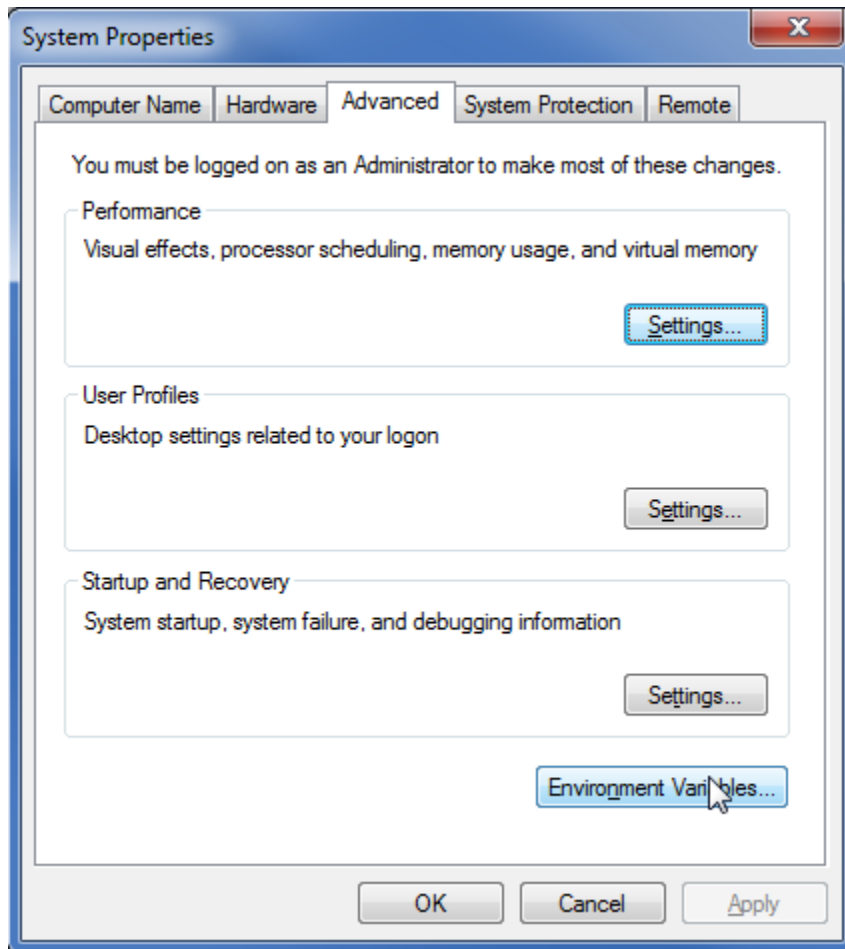
To use MATLAB® on a 32-bit or a 64-bit version of Microsoft Windows

- If the system has been integrated with MATLAB® previously, you do not have to restart the system since system variables have already been updated.
1. Install MATLAB®; restart MagicDraw.
  2. Press **Windows + R** to open the **Run** dialog.
  3. Type **cmd** in the open combo box and click **OK** to open the command prompt window.
  4. Type "matlab /regserver" and press **Enter** to register the MATLAB® components to Windows. The MATLAB® command prompt opens and is ready to use.

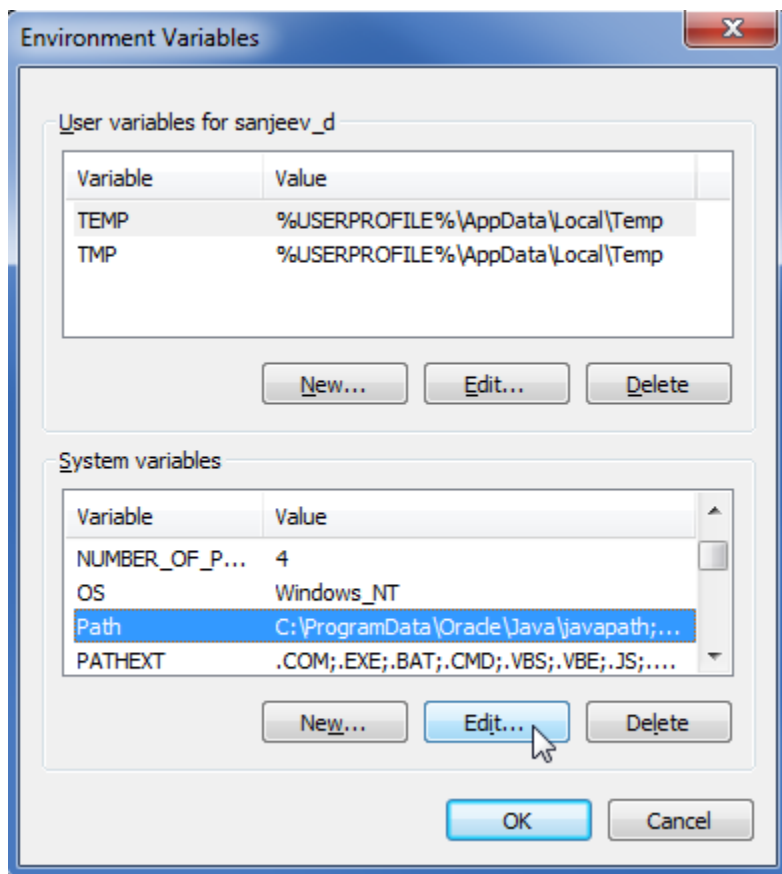


5. Add the path of the MATLAB® bin and bin/win32 (or bin/win64 for Microsoft Windows 64-bit) folders to the **Path** environment variable using the following steps

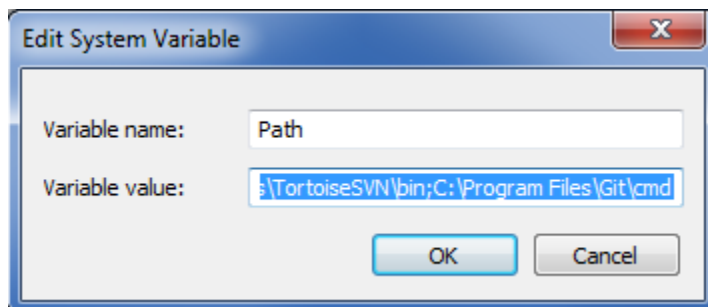
5.1 Double-click **System** in **Control Panel** to open the **System Properties** dialog. Click the **Advanced** tab.



5.2 Click **Environment Variables**. The **Environment Variables** dialog opens.



5.3 From the **System variables** list, select **Path** and click **Edit**. The **Edit System Variable** dialog opens.



5.4 In the **Variable value** box, enter the path to the MATLAB® bin and bin/win32 folders (or bin/win64 for Microsoft Windows 64-bit version), for example, ;C:\Program Files\MATLAB\R2010b\bin;C:\Program Files\MATLAB\R2010b\bin\win32;.

5.5 In the **Variable value** box, enter the path to the MATLAB® runtime/win32 (or runtime/win64 for Microsoft Windows 64-bit version), for example, ;C:\Program Files\MATLAB\R2010b\runtime\win32;.

**Note**  
You can skip Step 5.5 in the above procedure if the MATLAB® runtime directory does not exist, since this directory only existed in earlier versions of MATLAB® and not in the newer versions. The MATLAB® runtime is not required in these cases.

5.6 Click **OK**.

6. Restart Windows.

To use MATLAB® on Mac OS 10.6 (Snow Leopard)

1. Install MATLAB®.
2. Type the following commands in the terminal to show all files in Finder
  - \$ defaults write com.apple.finder AppleShowAllFiles TRUE
  - \$ killall Finder

3. Add the **DYLD\_LIBRARY\_PATH** variable to Mac OS:
  - 3.1 Create an empty text file in the `/etc` folder and name it: `launchd.conf`.
  - 3.2 Open it with a text editor, for example, TextEdit, and type the following text (no space)

```
setenv DYLD_LIBRARY_PATH /Applications/MATLAB_R2010b.app/bin/maci64:/Applications/MATLAB_R2010b.app/runtime/maci64
```

- 3.3 Save the text file as `launchd.conf` to the desktop.
- 3.4 Move the `launchd.conf` file to the `/etc` folder.

4. Create a link to the MATLAB® executable file in `/usr/bin` if it does not yet exist.
5. Call the following commands in the terminal
  - `$ cd /usr/bin`
  - `$ ln -s /Applications/MATLAB_R2010b.app/bin/matlab matlab`
6. Type the following commands in the terminal to reset Finder
  - `$ defaults write com.apple.finder AppleShowAllFiles FALSE`
  - `$ killall Finder`
7. Restart Mac OS.

**Note**  
You can also use MATLAB® if you are using MagicDraw 18.0 on either Mac OS X 10.10 Yosemite or Mac OS X 10.11 El Capitan, by using the following steps.

1. Disable the SIP (if your Mac is OS X El Capitan) by following the instruction [to disable OS X El Capitan's System Integrity Protection \(SIP\)](#). To use MATLAB® on 32-bit and 64-bit (tested with Ubuntu) versions of Linux
2. Install MATLAB®.

3. Create the file `com.nomagic.magicdraw.simulation.mathengine.plist`.
1. Install MATLAB® (Assume that your MATLAB installation directory is `/home/username/ MATHWORKS_ R2011A`).
2. Make sure that C Shell had already been installed on your Linux. To install C Shell on Ubuntu, type the following command in the terminal
  - `~$ sudo apt-get install csh`

```
3. Cr
• ~
• ~
• ~
• ~
4. Us
co
• on
/M
• on
/M
/M
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
<key>Label</key>
<string>com.nomagic.magicdraw.simulation.mathengine.plist</string>
<key>ProgramArguments</key>
<array>
<string>sh</string>
<string>-c</string>
<string>
launchctl setenv DYLD_LIBRARY_PATH
/Applications/MATLAB_R2012a.app/bin/maci64:/Applications/MATLAB_R2012a.app/runtime/maci64
</string></array><key>RunAtLoad</key><true/></dict></plist>
```

- 3.2 Change the `/Applications/MATLAB_R2012a.app` to your MATLAB directory.
- 3.3 Save the text file. (If you are using TextEdit, change the file to plain text by clicking **Format > Make Plain Text**).
- 3.4 Rename it as `com.nomagic.magicdraw.simulation.mathengine.plist`.

(Note: If you already have the file `com.nomagic.magicdraw.simulation.mathengine.plist` in `/Library/LaunchAgents`, add `:/Applications /MATLAB_R2012a.app/bin/maci64:/Applications/MATLAB_R2012a.app/runtime/maci64` to your `DYLD_LIBRARY_PATH` in your `com.nomagic.magicdraw.simulation.mathengine.plist`. For example, `launchctl setenv DYLD_LIBRARY_PATH <Other_Path>:/Applications/MATLAB_R2012a.app/bin/maci64:/Applications/MATLAB_R2012a.app/runtime/maci64`.)

4. Create the file `com.nomagic.magicdraw.simulation.mathengine.matlab.plist`.
  - 4.1 Create a text file and type the following text.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
<key>Label</key>
<string>com.nomagic.magicdraw.simulation.mathengine.matlab.plist</string>
<key>ProgramArguments</key>
<array>
<string>sh</string>
<string>-c</string>
<string>
launchctl setenv MD_MATLAB_MATHENGINE
/Applications/MATLAB_R2012a.app/bin/maci64: /Applications/MATLAB_R2012a.app/runtime/maci64
</string></array><key>RunAtLoad</key><true/></dict></plist>
```

```
cd "$APP_HOME"
```

4.2 Change "/Applications/MATLAB\_R2012a.app" to your MATLAB directory

4.3 Save the text file. (If you are using TextEdit, change the file to plain text by clicking **Format > Make Plain Text**).

4.4 Rename it as `com.nomagic.magicdraw.simulation.mathengine.matlab.plist`.

```
"$JAVA_LAUNCHER" "${JAVA_ARGS[@]}" "${MAC_ARGS[@]}" "$BOOT_CLASSPATH" "-
Launcher.properties.file=$PROP_FILE" "$MAIN_CLASS" "-cp" "$CLASSPATH"
com.nomagic.Launcher.Launcher "$APP_ARGS[@]" "$@"
```

5. Move the files to `/Library/LaunchAgents/` by using the Terminal.

5.1 Run the Terminal.

5.2 Go to the plist files directory.

```
echo "ERROR!"
echo "Java executable not found in" $JAVA_HOME/bin/java
echo "Please add java to your PATH environment variable, or specify it in
file" $PROP_FILE
```

5.3 Move the plist file to `/Library/LaunchAgents/` using the following command

- `$ sudo mv com.nomagic.magicdraw.simulation.mathengine.plist /Library/LaunchAgents/`
- `$ sudo mv com.nomagic.magicdraw.simulation.mathengine.matlab.plist /Library/LaunchAgents/`

## Related page

6. Create a link to the MATLAB® executable file in `/usr/bin` if it does not exist, by using the following command in the Terminal

- `$ cd /usr/bin`
- `$ sudo ln -s /Applications/MATLAB_R2012a.app/bin/matlab matlab`



## Note

You need to change the `/Applications/MATLAB_R2012a.app` in the command line to your MATLAB directory.

7. Restart Mac OS.