Pirana – RsNLME Installation

Table of Contents

RsNLME provides users with the ability to define PK/PD models from within the R environment and utilize PML (Phoenix Modeling Language) and the NLME engine to fit/simulate those models. Pirana is a flexible Pharmacometric Modeling Workbench and enhancements to Pirana 3.0 give users the ability to utilize Certara's NLME engine via RsNLME, alongside NONMEM.

Pirana Installation

Notes for Pirana 2.9.9 Users

If you have Pirana 2.9.9 installed:

- The uninstall wizard is Pirana 3.0-specific and will not even recognize Pirana 2.9.9.
- Installing Pirana 3.0 using the default directory (Pirana) will overwrite Pirana 2.9.9. Therefore, you should completely uninstall Pirana 2.9.9, including deleting the preferences directory located at `c:\users\<yourusername>\pirana` prior to installing 3.0. You can, however, specify an alternate directory (e.g., pirana30) to have parallel installations.

Windows (64-bit)

Download and install pirana_desktop_windows_3.0.0.exe.

1. Extract the ZIP file that contains the installation files.
2. Double-click installer executable.
3. In the *User Account Control* dialog, enter the username and password credentials for an administrator account.
4. On the Welcome screen, click **Next**.
5. On the Destination Folder screen, specify the location for the Pirana installation and click **Next**.
6. Click **Next** to start the install.
   The next screen shows the progress of the installation.
7. Once the installation is complete, click **Finish** to exit the installer.

Upon initial launch of a new installation on Windows, you may encounter a *Windows Security Alert* dialog regarding Windows Defender Firewall blocking some features of the Pirana executable. In the dialog, check the appropriate box(es) to allow Pirana to communicate on the network(s).

Linux

Download and install pirana_desktop_linux_3.0.0.tar.gz.

1. Unpack the archive file you received from Certara.
2. Execute `sudo yum install xterm`
3. Execute `install_modules_linux.sh` to install dependencies.

Note: install_modules_linux.sh contains the core dependencies you need to install Pirana on Linux. Depending on your system environment, you may need to install additional modules, which is done using the following syntax:

```
sudo cpan install Module::Name
```

SSH software packages are required for remote execution and included in CentOS by default. If these packages are not present on your system, please see the following link for instructions on "How to Install/Enable OpenSSH on CentOS 7."

Install the Pirana license file

1. In Pirana, select **Help > Install license file**.
2. In the dialog, enter the path to the license file (click the folder icon to browse to the location) and click **Install license**.

The R Environment Installation

R ≥ 4.0.2 is required to run RsNLME from within Pirana. If you will be using standalone RsNLME in R (without Pirana), then RStudio > 1.2 is a recommended IDE.

If either R ≥ 4.0.2 or RStudio >1.2 is already installed, skip to the next section ("Certara R Packages – RsNLME").

R

Follow the links below to download R 4.0+ for your operating system and access the installation instructions.

- R 4.0.2+ for Windows
- R 4.0.2+ for Linux (***Note:*** You must select Linux distribution from the available options.)

Proceed to install R.

RStudio

Follow the link below to access the RStudio Desktop download.

1. RStudio Desktop

Note that, due to a known bug, RStudio 1.2 has some issues with the R "utils::file.edit()" function used for editing textual models. Updating to the latest version is recommended in order to directly edit PML statements for textual models within R Studio.

Proceed to install RStudio Desktop.

Certara R Packages – RsNLME

Execute the following commands from R commandline to install Certara R packages. There are a number of packages that may need to be installed, so it may take several minutes for the command execution to complete.

Windows

```
Certara_Packages <- c("Certara.RsNLME", "Certara.NLME8",
"Certara.Xpose.NLME", "Certara.RsNLME.ModelExecutor",
"Certara.RsNLME.ModelBuilder")


install.packages(Certara_Packages, repos =
c("https://certara.jfrog.io/artifactory/certara-cran-release-public/",
"https://cloud.r-project.org"), method = "libcurl")
```

Linux

RsNLME requires the "XML" dependency package from R. In order to successfully install all `Certara_Packages` (which will automatically install the "XML" dependency), you may need to first run the following command from your Linux terminal.

```
sudo yum install libxml2-devel
```

Then proceed to install `Certara_Packages` from R.

```
Certara_Packages <- c("Certara.RsNLME", "Certara.NLME8",
"Certara.Xpose.NLME", "Certara.RsNLME.ModelExecutor",
"Certara.RsNLME.ModelBuilder")


install.packages(Certara_Packages, repos =
c("https://certara.jfrog.io/artifactory/certara-cran-release-public/",
"https://cloud.r-project.org"))
```

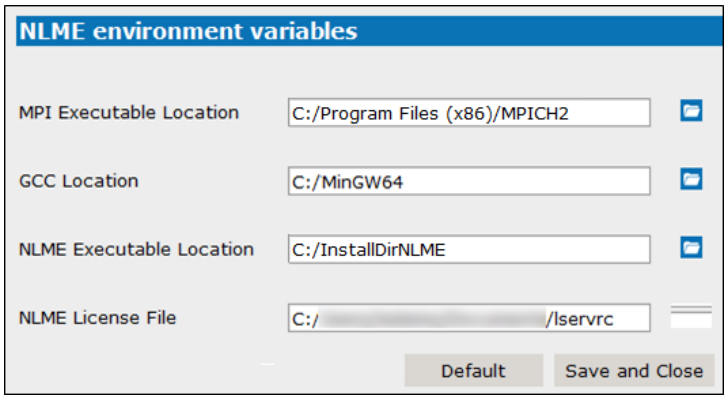
NLME Executables

To prepare NLME executables for use, unzip the NLME executables zip file you received from Certara (`Certara_Nlme_Executables_8.4.0.zip`) in a directory of your choice. It unzips to a folder called `InstallDirNLME`. Make a note of the location, you will need this information to configure RStudio and/or Pirana to successfully run NLME models.

Environment Variables and Licensing for NLME

Pirana Settings

1. In Pirana, select **File > Settings** to display the *Pirana Settings* dialog.
2. Select **NLME** in the list on the left.



3. Specify the **MPI Executable Location** in the field.
4. Specify the **GCC Location** in the field.
5. Specify the **NLME Executable Location** in the field.
6. Specify the **NLME License File** in the field and click **Save and Close**.

NLME Executables

Create an environment variable *INSTALLDIR* using one of the following methods.

1. Create an environment variable for the current R session in RStudio, use the command below.

   ```
   Sys.setenv("INSTALLDIR"= "C:\\InstallDirNLME\\")
   ```
   (Be sure to edit the above command with the appropriate path first.)

2. Create a permanent environment variable on the host where the jobs will be executed.

   Windows

   Open the environment variable window using the following command and add the *INSTALLDIR* variable there.

   ```
   start powershell -command "&{start-process rundll32.exe
   sysdm.cpl,EditEnvironmentVariables -verb RunAs}"
   ```

   Or

   Run in cmd with admin privileges:

   ```
   setx INSTALLDIR=C:\\InstallDirNLME\\
   ```
   (Be sure to edit the above command with the appropriate path first.)

Linux

Edit .bashrc in your home directory:

```
echo 'export INSTALLDIR='~/InstallDirNLME/' >> ~/.bashrc
```

(Be sure to edit the above command with the appropriate path first.)

Execute `initpml`.

```
sudo ./initpml
```

(Be sure you are inside `InstallDirNLME` folder when executing above command, i.e., `cd ./InstallDirNLME`*)*

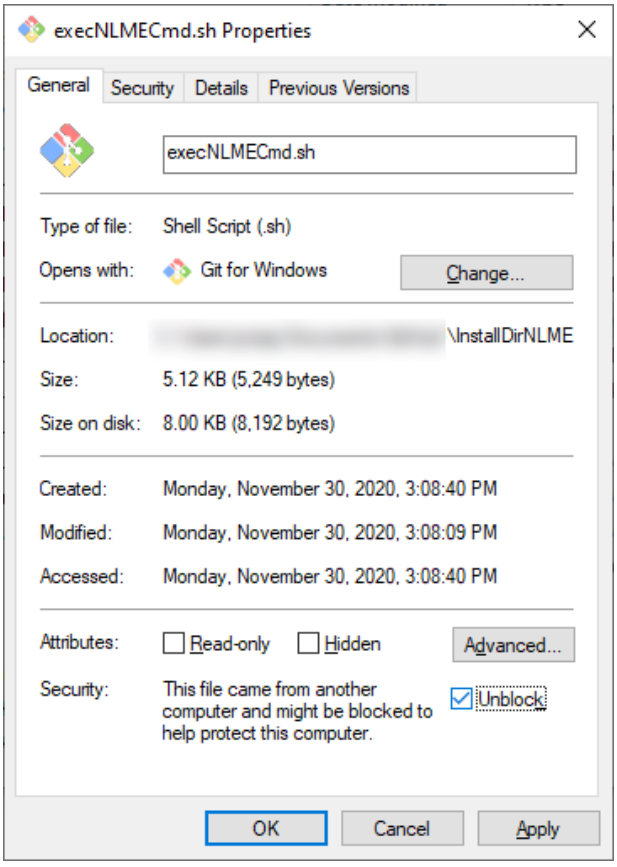
Note: Depending on the security settings of your organization, some files inside InstallDirNLME may need to be unblocked. Please make sure the following files in InstallDirNLME have appropriate permissions by right-clicking filename, selecting **Properties**, and checking the **Unblock** box.

Windows

```
execNLMECmd.ps1
common.ps1
```

Linux

```
execNLMECmd.sh
```



MPI Location

Windows – MPICH2 on Windows

Installation

Skip this installation section if Phoenix with MPICH is already installed.

1. Open the file `mpich2-1.4.1p1-win-x86-64.msi`, which you received from Certara and is in your installation folder.
2. Follow the installer instructions to install MPICH2.

Set environment variable

Create the PhoenixMPIDir64 environment variable using one of the following methods.

1. Create an environment variable for the current R session (by default, it is installed into *C:\Program Files\MPICH2*).

   ```
   Sys.setenv("PhoenixMPIDir64"= "C:\\Program Files\\MPICH2\\")
   ```
   (Be sure to edit the above command with the appropriate path first.)

2. Open the environment variable window using the following command and add the *PhoenixMPIDir64* variable there.

   ```
   start powershell -command "&{start-process rundll32.exe
   sysdm.cpl,EditEnvironmentVariables -verb RunAs}"
   ```

   Or

   Run in cmd with admin privileges:

   ```
   setx PhoenixMPIDir64=C:\\ProgramFiles\\MPICH2\\
   ```
   (Be sure to edit the above command with the appropriate path first.)

To install and start SMPD service on the system, use the following command with Admin privileges: `c:\Program Files\MPICH2\bin\smpd.exe –install`

SMPD will attempt to find slave nodes outside of the local host and that action is caught by Windows Firewall as an annoying pop-up dialog prompting to allow mpiexec to connect to the other nodes.

If you experience issues with Windows Firewall, and to prevent the pop-up, execute the `change_firewall_rules` script that you received in your download materials from Certara, as an admin.

Linux – OpenMPI

On a Linux platform, the MPI implementation supported is OpenMPI (version 1.10 or higher). For installation instructions, refer to the OpenMPI site.

After installation, make sure to set PATH and LD_LIBRARY_PATH properly.

Note: If the MPI (not local_mpi) host is intended to be in use, the Rmpi package should be installed on the host machine. For instructions see here.

GCC Location

Windows

NLME executables were extensively tested within MinGW GCC v.4.9.2 rev.0. It is highly recommended to use that version to avoid unpredicted behavior. Older versions of MinGW GCC are not supported. Some versions after 4.9.2 may work, but warnings will be given.

Installation

Skip this installation section if Phoenix with GCC MinGW is already installed.

1. Unzip the file `x86-64-4.9.2-release-posix-seh-rt_v3-rev0.7z`, which you received from Certara and is in your installation folder.
2. Open the unzipped `x86-64-4.9.2-release-posix-seh-rt_v3` folder copy the **mingw64** sub-folder to a location you will remember (e.g., `C:\\mingw64`).

Set environment variable

Create the *PhoenixGCCDir64* environment variable using one of the following methods.

1. Create an environment variable for the current R session.

   ```
   Sys.setenv("PhoenixGCCDir64"= "C:\\PHSTMinGW64\\")
   ```
   (Be sure to edit the above command with the appropriate path first.)

2. Open the environment variable window using the following command and add the *PhoenixGCCDir64 variable there.*

   ```
   start powershell -command "&{start-process rundll32.exe
   sysdm.cpl,EditEnvironmentVariables -verb RunAs}"
   ```

   Or

   Run in cmd with admin privileges:

   ```
   setx PhoenixGCCDir64=C:\\PHSTMinGW64\\
   ```
   (Be sure to edit the above command with the appropriate path first.)

Linux

GCC version 4.8.5 is supported (distributed within Red Hat Enterprise Linux 7 and alternatives). Older versions of GCC are not supported. Some versions after 4.8.5 could work, but warnings may be given.

NLME Working Directory

NLME working directory environment variable only needs to be explicitly set in RStudio.

Identify the directory where the working folders will be created. This can be done using one of the following options:

1. *Suggested*. Create an environment variable for the current R session.

   ```
   Sys.setenv("NLME_ROOT_DIRECTORY"= getwd())
   ```
   File path can be any valid directory, or the user's current directory in R using `getwd()`.

2. Edit NLME host class instance.

See `?Certara.NLME8::NlmeParallelHost` and examples after corresponding package installation.

***Note**:* NLME host class instance slot has higher priority than the *NLME_ROOT_DIRECTORY* environment variable.

Setting NLME License

An NLME license is required to execute models in RsNLME via the Certara.NLME8 package. If you do not currently have an NLME license, please contact licensing@certara.com. The license file should be named `lservrc` and contain a license called "*NU_PMLCommandLine_N*".

When using Pirana, identify the location of the license file as follows:

1. In Pirana, select **File > Settings** to display the *Pirana Settings* dialog.
2. Select **NLME** in the list on the left.
3. Specify the full path to the `lservrc` file in the **NLME License File** field and click **Save and Close**.

When using RStudio, create an environment variable to define the license file, for local execution, or the license server, for remote execution.

1. *Local Execution*: License file with NU_PMLCommandLine license *PhoenixLicenseFile* permanent or session-specific environment variable should be provided:

   ```
   Sys.setenv("PhoenixLicenseFile" = "C:\\lservrc")
   ```
   (Edit above path before running.)

2. *Remote Execution:* Named user license NUS_PMLCommandLine *PhoenixLicenseServer* permanent or session-specific environment variable should be provided:

   ```
   Sys.setenv("PhoenixLiceseServer"= 10.11.12.13)
   ```
   (Edit server location before running.)

Note: *PhoenixLiceseServer* has higher precedence than *PhoenixLicenseFile*.