

NLME Standalone User's Guide

**Applies to:
Certara.NLME Package 8.0**

NLME Standalone Version 8.0

Phoenix® WinNonlin®, Phoenix NLME™, IVIVC Toolkit™, CDISC® Navigator, AutoPilot Toolkit™, Job Management System™ (JMS™), Pharsight Knowledgebase Server™ (PKS™), Trial Simulator™, Validation Suite™ copyright ©2005-2018, Certara, L.P. All rights reserved. This software and the accompanying documentation are owned by Certara, L.P. The software and the accompanying documentation may be used only as authorized in the license agreement controlling such use. No part of this software or the accompanying documentation may be reproduced, transmitted, or translated, in any form or by any means, electronic, mechanical, manual, optical, or otherwise, except as expressly provided by the license agreement or with the prior written permission of Certara, L.P.

This product may contain the following software that is provided to Certara, L.P. under license: Formula One® copyright 1993-2018 Open-Text Corporation. All rights reserved. Microsoft® .NET Framework copyright 2018 Microsoft Corporation. All rights reserved. Tab Pro ActiveX 2.0.0.45 copyright 1996-2018, GrapeCity, Inc. All rights reserved. Sentinel™ RMS 8.4.0.900 copyright 2006-2018 Gemalto NV. All rights reserved. Microsoft XML Parser version 3.0 copyright 1998-2018 Microsoft Corporation. All rights reserved. Websites Screenshot DLL 1.6 copyright 2008-2018 WebsitesScreenshot.com. All rights reserved. Certara, L.P. has agreement to use and redistribute licenses for the following software: Syncfusion Essential Studio Enterprise 15.4.0.17 copyright 2001-2018 Syncfusion Inc. All rights reserved. Minimal Gnu for Windows (MinGW, <http://mingw.org/>), copyright 2004-2018 Free Software Foundation, Inc. This product may also contain the following royalty free software: DotNetbar 1.0.0.24030 (with custom code changes) copyright 1996-2018 DevComponents LLC. All rights reserved. Xceed zip Library 2.0.116.0 copyright 1994-2018 Xceed Software Inc. All rights reserved. IMSL® copyright 2018 Rogue Wave Software, Inc. All rights reserved.

Information in the documentation is subject to change without notice and does not represent a commitment on the part of Certara, L.P. The documentation contains information proprietary to Certara, L.P. and is for use by its affiliates' and designates' customers only. Use of the information contained in the documentation for any purpose other than that for which it is intended is not authorized. NONE OF CERTARA, L.P., NOR ANY OF THE CONTRIBUTORS TO THIS DOCUMENT MAKES ANY REPRESENTATION OR WARRANTY, NOR SHALL ANY WARRANTY BE IMPLIED, AS TO THE COMPLETENESS, ACCURACY, OR USEFULNESS OF THE INFORMATION CONTAINED IN THIS DOCUMENT, NOR DO THEY ASSUME ANY RESPONSIBILITY FOR LIABILITY OR DAMAGE OF ANY KIND WHICH MAY RESULT FROM THE USE OF SUCH INFORMATION.

Destination Control Statement

All technical data contained in the documentation are subject to the export control laws of the United States of America. Disclosure to nationals of other countries may violate such laws. It is the reader's responsibility to determine the applicable regulations and to comply with them.

United States Government Rights

This software and accompanying documentation constitute “commercial computer software” and “commercial computer software documentation” as such terms are used in 48 CFR 12.212 (Sept. 1995). United States Government end users acquire the Software under the following terms: (i) for acquisition by or on behalf of civilian agencies, consistent with the policy set forth in 48 CFR 12.212 (Sept. 1995); or (ii) for acquisition by or on behalf of units of the Department of Defense, consistent with the policies set forth in 48 CFR 227.7202-1 (June 1995) and 227.7202-3 (June 1995). The manufacturer is Certara, L.P., 100 Overlook Center, Suite 101, Princeton, New Jersey, 08540.

Trademarks

AutoPilot Toolkit, IVIVC Toolkit, Job Management System, JMS, NLME, Pharsight Knowledgebase Server, PKS, Phoenix, Trial Simulator, Validation Suite, WinNonlin are trademarks or registered trademarks of Certara, L.P. NONMEM is a registered trademark of ICON Development Solutions. S-PLUS is a registered trademark of Insightful Corporation. SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. Sentinel RMS is a trademark of Gemalto NV. Microsoft, MS, .NET, SQL Server Compact Edition, the Internet Explorer logo, the Office logo, Microsoft Word, Microsoft Excel, Microsoft PowerPoint®, Windows, the Windows logo, the Windows Start logo, and the XL design (the Microsoft Excel logo) are trademarks or registered trademarks of Microsoft Corporation. Pentium 4 and Core 2 are trademarks or registered trademarks of Intel Corporation. Adobe, Acrobat, Acrobat Reader, and the Adobe PDF logo are registered trademarks of Adobe Systems Incorporated. All other brand or product names mentioned in this documentation are trademarks or registered trademarks of their respective companies or organizations.

Additional third party software acknowledgements

Software for Locally-Weighted Regression

The authors of this software are Cleveland, Grosse, and Shyu. Copyright © 1989, 1992 by AT&T. Permission to use, copy, modify, and distribute this software for any purpose without fee is hereby granted, provided that this entire notice is included in all copies of any software which is or includes a copy or modification of this software and in all copies of the supporting documentation for such software.

This software is being provided “as is”, without any express or implied warranty. In particular, neither the authors nor AT&T make any representation or warranty or any kind concerning the merchantability of this software or its fitness for any particular purpose.

LAPACK

Copyright © 1992-2007 The University of Tennessee. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer listed in this license in the documentation and/or other materials provided with the distribution.

Neither the name of the copyright holders nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

This software is provided by the copyright holders and contributors “as is” and any express or implied warranties, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose are disclaimed. In no event shall the copyright owner or contributors be liable for any direct, indirect, incidental, special, exemplary, or consequential damages (including, but not limited to, procurement of substitute goods or services; loss of use, data, or profits; or business interruption) however caused and on any theory of liability, whether in contract, strict liability, or tort (including negligence or otherwise) arising in any way out of the use of this software, even if advised of the possibility of such damage.

NLog

Copyright © 2004-2006 Jaroslaw Kowalski <jaak@jkowalski.net>. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of Jaroslaw Kowalski nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

This software is provided by the copyright holders and contributors “as is” and any express or implied warranties, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose are disclaimed. In no event shall the copyright owner or contributors be liable for any direct, indirect, incidental, special, exemplary, or consequential damages (including, but not limited to, procurement of substitute goods or services; loss of use, data, or profits; or business interruption) however caused and on any theory of liability, whether in contract, strict liability, or tort (including negligence or otherwise) arising in any way out of the use of this software, even if advised of the possibility of such damage.

Contents

Chapter 1	Introduction	7
	Certara contact information	7
Chapter 2	NLME Standalone Installation	9
	Linux instructions	9
	Installation steps	9
	Test installation	11
	Validation	12
	Windows instructions	12
	Installation steps	12
	Test Installation	14
	Validation	14
Chapter 3	NLME Standalone Command Line Help	17
Chapter 4	NLME Standalone Examples	19

Introduction

The Certara.NLME R package and its supporting tools are shipped with the Phoenix installer and enable the execution of NLME in parallel on remote/local hosts from the Phoenix graphical user interface. When the package is installed, an installation directory is created on the requested host, remote or local, that includes all platform-specific executables, libraries, source and scripts needed to execute NLME jobs in parallel.

This package is also available as a standalone product and can be used from command line independent of Phoenix.

The *NLME Standalone User's Guide* contains the following topics:

- “Certara contact information” on page 7
- “NLME Standalone Installation” on page 9
- “NLME Standalone Command Line Help” on page 17

Certara contact information

Technical Support

Consult the software documentation to address questions. If further assistance is needed, contact Certara Support through e-mail or our web site.

E-mail: support@certara.com

Web: support.certara.com/support

For the most efficient service, e-mail a complete description of the problem, including copies of the input data.

User Forum

Get tips and discuss Certara software with other users at:

<https://support.certara.com/forums>

NLME Standalone Installation

For Linux installation instructions, see [“Linux instructions” on page 9](#)

For Windows installation instructions, see [“Windows instructions” on page 12](#).

The NLME Standalone installation zip/tar file contains the following:

- **InstallDirNLME directory:** This is the NLME installation directory containing all platform-specific shell scripts, templates, libraries, and executables to run NLME. It also contains the Certara.NLME8.x... package.
- **Examples:** For each NLME run mode, there is a directory that has all the data/model files necessary plus an R script, a Windows batch file or Linux shell script example to run that particular mode of NLME (run.r, run.bat, run.sh).
 - **bootstrap:** Performs a Bootstrap execution.
 - **generic_run:** Performs a Simple estimation and VPC Simulation
 - **profile_estimation:** Performs a Profile Perturbation
 - **shotgun_covarsrch:** Performs a Shotgun Covariate Search
 - **sortcol_estimation:** Performs a Multiple Estimation with Sort column enabled
 - **stepwise_covarsrch:** Performs a Stepwise Covariate Search
- NLME Standalone.pdf help file.

Linux instructions

Installation steps

1. Acquire the Certara_Nlme_Linux.tar file from Certara.
2. Log into your Linux server and copy the package to a directory (referred to as `/home/nlmeuser/Work` in this document).
3. Type following command:

```
tar -xvf /home/nlmeuser/Work/Certara_Nlme_Linux.tar
```
4. If this is the first time installation, install the required R packages:
 - XML
 - reshape
 - batchtools
5. Install/Update the Certara.NLME R library by typing following commands:

```
cd InstallDirNLME
sudo R CMD INSTALL Certara.NLME8_0.0.1.0001.tar.gz
```

The following information is displayed during the installation process:

```
[sudo] password for nlmeuser:
* installing to library '/usr/lib64/R/library'
* installing *source* package 'Certara.NLME8' ...
** R
** preparing package for lazy loading
** help
*** installing help indices
    converting help for package 'Certara.NLME8'
      finding HTML links ... done
      performBootstrap                        html
      performEstimationOnSortColumns          html
      performParallelNLMERun                  html
      performShotgunCovarSearch               html
      performStepwiseCovarSearch              html
      reconnectToBootstrapNLMERun             html
      summarizeBootstrap                      html
** building package indices
** testing if installed package can be loaded
* DONE (Certara.NLME8)
Making 'packages.html' ... done
```

6. Set the NLME_ROOT_DIRECTORY environment variable by typing:

```
Export NLME_ROOT_DIRECTORY=<directory path>
```

7. Set the PhoenixLicenseServer variable by typing:

```
Export PhoenixLicenseServer==<licenser server>
```

8. Customize the installation for R usage.

A site customization R script, `setup_env.r`, is placed during installation in the `InstallDirNLME` directory. Users can edit this file to reflect site-specific variables and load required R libraries. When the `Certara.NLME` package is loaded, it will check for `.../InstallDirNLME/setup_env.r` and will source in this script, if it exists.

The contents of the default `setup_env.r` script supplied with the package are shown below:

```
library(parallel)
library(XML)
#
# Check environment variables
#
rootDirectory=Sys.getenv("NLME_ROOT_DIRECTORY")
if ( rootDirectory == "" )
{
  warning("Environment variable NLME_ROOT_DIRECTORY is not defined,
skipping initialization")
  stop()
}
#
# Windows long path fix#
#
if ( Sys.info()["sysname"] == "Windows" )
  rootDirectory=shortPathName(rootDirectory)
#
# Setup environment variables
#
Sys.setenv("INSTALLDIR"=paste0(rootDirectory,"/InstallDirNLME"))
#
# Example of a parallel host for Multicore execution
#
host1 = NlmeParallelHost(sharedDirectory=rootDirectory,
```

```

parallelMethod=NlmeParallelMethod("MULTICORE"),
hostName="Multicore",
numCores=4)

#
# Example of a TORQUE_MPI parallel host
#
host2 = NlmeParallelHost(sharedDirectory=rootDirectory,
                        parallelMethod=NlmeParallelMethod("TORQUE_MPI"),
                        hostName="TorqueGrid",
                        numCores=32)

#
# Example of an SGE_MPI grid
#
host3 = NlmeParallelHost(sharedDirectory=rootDirectory,
                        parallelMethod=NlmeParallelMethod("SGE_MPI"),
                        hostName="SgeGrid",
                        numCores=8)

host4 = NlmeParallelHost(sharedDirectory=rootDirectory,
                        parallelMethod=NlmeParallelMethod("LOCAL_MPI"),
                        hostName="LOCAL_MPI",
                        numCores=4)

#
# Make example hosts available
#
hosts=c(host1,host2,host3,host4)
#
# Some defaults
#
defaultParams=NlmeEngineExtraParams()
defaultDataset=NlmeDataset()
defaultHost=hosts[[1]]

```

Test installation

1. Choose one example and go to the examples directory using the command:

```
cd /home/nlmeuser/Work/Examples/Bootstrap
```

2. Run the following command to open the file in the VM editor:

```
vim run.sh
```

3. Adjust the command line to reflect the current parallel environment.

The example script is set to use a TORQUE_MPI environment with 32 cores. This needs to be adjusted to reflect the platform you are using by modifying the underlined text.

```

${INSTALLDIR}/bootstrap.sh TORQUE_MPI ${INSTALLDIR} ${shared_directory}
`pwd` 3 1000 10 3 test.mdl cols1.txt data1.txt 11938 nlmeargs.txt
"nlmeargs.txt test.mdl nlmeargs.txt cols1.txt data1.txt test.mdl" 32 95
TRUE

```

4. Run the following command:

```
./run.sh
```

Note: If you have problems starting the job, try the following:

Set the root by typing `export NLME_ROOT_DIRECTORY=/home/nlmeuser/Work`

Set the license by typing `export PhoenixLicenseServer=<licenseserver>`
or by typing `export NLME_HASH=<hash_code_from_Certara>`

Check that the root has been set by typing `env | grep ROOT`

Validation

1. In Bootstrap folder set command:

```
ls -l    (small L)
```

2. Check that there are following files:

BootOmega.csv	EtaEta.txt
BootOmegaStacked.csv	EtaShrinkageBySubject.txt
BootOmegaStderr.csv	IdEta.txt
BootOverall.csv	IniCovr.txt
BootSubj.csv	MultCovr.txt
BootTheta.csv	nlmeargs.txt
BootThetaStacked.csv	out.bin.txt
BootVarCoVar.csv	out_initialEstimates.txt
cols1.txt	out.txt
data1.txt	progress.txt
dmp.txt	run.sh
err2.txt	StrCov.txt
EtaCov.txt	test.mdl

Windows instructions

Installation steps

1. Acquire the Certara_Nlme_Windows.zip file from Certara.
2. Unzip the file into a directory (referred to as *Work* in this document).
3. Run Windows command line.
4. If this is the first time installation, install the required R packages:
 - XML
 - reshape
 - batchtools
5. Install/Update the Certara.NLME R library by typing following commands:

```
cd C:\Work\TestInstallNLME\InstallDirNLME
```

```
"C:\Program Files\R\R-3.2.2\bin\R.exe" R CMD INSTALL Cer-  
tara.NLME8_0.0.1.0001.tar.gz
```

The following information is displayed during the installation process:

```
* installing to library '/usr/lib64/R/library'
* installing *source* package 'Certara.NLME8' ...
** R
** preparing package for lazy loading
** help
```

```

*** installing help indices
** building package indices
** testing if installed package can be loaded
* DONE (Certara.NLME8)

```

6. Set the NLME_ROOT_DIRECTORY environment variable by typing:

```
setenv NLME_ROOT_DIRECTORY "<directory path>"
```

7. Set the PhoenixLicenseServer variable by typing:

```
setenv PhoenixLicenseServer "<licenser server>"
```

If this variable is not set, the program will look for the Phoenix license installation file in :

```
C:\Program Files (x86)\Certara\Phoenix\application\Services\Licensing\lsrvrc
```

Alternatively, you can point to a license file in a different location by providing the full path to license file in PhoenixLicenseFile environment variable:

```
setenv PhoenixLicenseFile "<full path to license file>"
```

8. Customize the installation for R usage.

A site customization R script, `setup_env.r`, is placed during installation in the `Install-DirNLME` directory. Users can edit this file to reflect site-specific variables and load required R libraries. When the Certara.NLME package is loaded, it will check for `.../InstallDirNLME/setup_env.r` and will source in this script, if it exists.

The contents of the default `setup_env.r` script supplied with the package are shown below:

```

library(parallel)
library(XML)
#
# Check environment variables
#
rootDirectory=Sys.getenv("NLME_ROOT_DIRECTORY")
if ( rootDirectory == "" )
{
  warning("Environment variable NLME_ROOT_DIRECTORY is not defined,
skipping initialization")
  stop()
}
#
# Windows long path fix#
#
if ( Sys.info()["sysname"] == "Windows" )
  rootDirectory=shortPathName(rootDirectory)
#
# Setup environment variables
#
Sys.setenv("INSTALLDIR"=paste0(rootDirectory,"/InstallDirNLME"))
#
# Example of a parallel host for Multicore execution
#
host1 = NlmeParallelHost(sharedDirectory=rootDirectory,
                        parallelMethod=NlmeParallelMethod("MULTICORE"),
                        hostName="Multicore",
                        numCores=4)
#
# Example of a TORQUE_MPI parallel host
#
host2 = NlmeParallelHost(sharedDirectory=rootDirectory,
                        parallelMethod=NlmeParallelMethod("TORQUE_MPI"),
                        hostName="TorqueGrid",
                        numCores=32)
#

```

```
# Example of an SGE_MPI grid
#
host3 = NlmeParallelHost(sharedDirectory=rootDirectory,
                        parallelMethod=NlmeParallelMethod("SGE_MPI"),
                        hostName="SgeGrid",
                        numCores=8)
host4 = NlmeParallelHost(sharedDirectory=rootDirectory,
                        parallelMethod=NlmeParallelMethod("LOCAL_MPI"),
                        hostName="LOCAL_MPI",
                        numCores=4)

#
# Make example hosts available
#
hosts=c(host1,host2,host3,host4)
#
# Some defaults
#
defaultParams=NlmeEngineExtraParams()
defaultDataset=NlmeDataset()
defaultHost=hosts[[1]]
```

Note: The LOCAL_MPI platform (host4 in the `setup_env.r` file) is only supported on Windows. Simple fitting jobs benefit from this mode, whereas, other run modes (with multiple replicates) are faster using MULTICORE on Windows).

Test Installation

1. Choose one example and go to examples directory using command:

```
cd C:\Work\TestInstallNLME\Examples\Bootstrap
```

2. Run the command:

```
run.bat > test.txt 2>&1
```

Note: If you have problems starting the job, try the following:

Set the root by typing `set NLME_ROOT_DIRECTORY=C:\Work\TestInstallNLME\`

Set the license by typing `set PhoenixLicenseServer=<licenseserver>`
or by typing `set NLME_HASH=<hash_code_from_Certara>`

Validation

1. Open Bootstrap folder `C:\Work\TestInstallNLME\Examples\Bootstrap`
2. Check that there are following files:

BootOmega.csv	EtaShrinkageBySubject.txt
BootOmegaStacked.csv	IdEta.txt
BootOmegaStderr.csv	IniCovr.txt
BootOverall.csv	MultCovr.txt
BootSubj.csv	nlmeargs.txt
BootTheta.csv	out.bin.txt
BootThetaStacked.csv	out_initialEstimates.txt

BootVarCoVar.csv	out.txt
cols1.txt	progress.txt
data1.txt	run.bat
dmp.txt	run.r
err2.txt	run.sh
EtaCov.txt	StrCov.txt
EtaEta.txt	test.mdl

NLME Standalone Command Line Help

Descriptions of the available arguments for each of the NLME Standalone command line scripts can be accessed by invoking the script without arguments.

For example, from the InstallDirNLME directory, type:

Linux: `./bootstrap.sh`

Windows Command Prompt: `bootstrap.bat`

A description of the bootstrap command and its arguments, similar to the following, is displayed in the window.

Usage: `bootstrap.sh <bootstrap.bat for Windows>`

Arg1 = PARALLEL_MECHANISM
NONE|MULTICORE|SGE|SGE_MPI|TORQUE|TORQUE_MPI|LSF|LSF_MPI

Arg2 = INSTALLATION_DIRECTORY
Contains NLME libraries and shell scripts for the platform.

Arg3 = SHARED_DIRECTORY
Use this location to create temporary working directories.

Arg4 = LOCAL_DIRECTORY
Input and results files are expected/placed here.

Arg5 = METHOD
1 = QRPEM
2 = IT2S-EM
3 = FOCE L-B
4 = First Order
5 = FOCE ELS
5 = Laplacian
6 = Naive pooled

Arg6 = ITERATIONS
Maximum number of iterations to perform.

Arg7 = NUM_SAMPLES
Number of bootstrap replicates to run.

Arg8 = NUM_RETRIES
Maximum number of retries for each replicate.

Arg9 = MODEL_FILE
Text file containing the PML model.

Arg10 = COL_DEF_FILE
Mapping between model and data file column names.

Arg11 = DATA_FILE
Text file containing the observations.

Arg12 = START_SEED
Random number generator seed.

Arg13 = EXTRA_ARGS_FILE
Text file containing NLME engine options.

Arg14 = FILES_TO_COPY
Space-separated list of files to copy to a remote directory
(e.g., test.mdl data1.txt colsl.txt nlmeargs.txt).

Arg15 = NUM_PROCESSES
Number of processes to use for parallelization
(e.g., number of cores on the remote host or nodes on a grid).

Arg16 = CONFIDENCE_LEVEL
Confidence level for summarization (e.g., 95).

NLME Standalone Examples

The Examples directory installed with NLME Standalone contains folders with the files needed to execute the different NLME run modes in R.

The following files are found in these folders:

- col(1,2,...).txt: Mapping of columns.
- covariates.txt: Mapping and initial estimates of the covariates.
- data(1,2,...).txt: Columns of input data.
- nlmeargs.txt or nlmeargsCombined.txt: Contains the NLME job settings, most of which correspond to the options available in the Run Options tab of the NLME object in Phoenix.
- nlmeControlFile.txt: Contains model specifications, such as the name of the data file, column mapping file, output files, and engine type selection.
- predckargs.txt: For Predictive Check runs. The same as the nlmeargs.txt file, except it also contains the predictive check-specific settings.
- run.bat: Windows batch file that contains the information required to run the NLME job on a Windows machine. (Examples/Simple also has runmpi.bat, which illustrates modifying run.bat to use the LOCAL_MPI parallel platform.)
- run.r: R script that contains the information required to run the NLME job using the R program. (Examples/Simple and Simple2 also have runmpi.r, which illustrates modifying run.r to use the LOCAL_MPI parallel platform.)
- run.sh: Linux shell script that contains the information required to run the NLME job on a Linux machine.
- test.mdl: The NLME model to be run.

Looking more closely at the Bootstrap example files, the input data in the data1.txt file includes xid, dose, time, yobs, and wt columns of values. These columns are mapped to id, time, dose, obs (CObs), and covr (wt) contexts using the cols1.txt file.

The test.mdl file defines a single compartment PK clearance model.

The NLME arguments are taken from the nlmeargs.txt file. The file begins with defining the engine (/m 3 which indicates the FOCE L-B engine), number of iterations (/n 1000), ODE level (/o 6, matrix exponent), and whether to enable fixed effects (/e -1, no). Next is a list of the arguments:

- xnp 0 – A value of 0 means do not run, otherwise enter the number of nonparametric generations
- anagrad 0 – A value of 0 means that synthetic gradients will not be used, 1 means, in population modeling, the engine computes and makes use of analytic gradients with respect to etas.
- logtran 1 – A value of 1 log-transforms the predictions and observations, 0 means no log-transformation is performed.

- xrestart 0 – A value of 0 means use the starting values in the model file as the initial solution source, 1 means read the initial solution from the restart file created during a previous run of the same model
- xnorderagg 1 – This argument is only used with engine 5. Engine 3 is used for this particular example, so the argument will be ignored. A value of 1 will use Laplacian approximation. A value >1 indicates the number of Adaptive Gaussian Quadrature (AGQ) points along each random effects dimension.
- xfocehess 1 – A value of 1 will use FOCE to approximate the Hessian matrix, a value of 0 will use numerical differentiation.
- xverbose 1 – Verbose mode is always used.
- xpresample 0 — This argument is not currently used.
- xmapnp 0 – A value of 0 will not perform any MAP_NP iterations, otherwise enter the number of MAP_NP iterations to run.
- xstderr 1 – A value of 1 indicates central diff, 0 means no standard error computation, 2 indicates forward diff.
- xlameth 1 – This argument is only used with engine 5. Engine 3 is used for this particular example, so the argument will be ignored. A value of 1 will use line search as the method for optimizing the overall likelihood function, 2 will use dogleg, 3 will use Levenberg-like trust region.
- xlandig 7 – Estimate of the available accuracy of the optimized overall likelihood function in terms of decimal places.
- xlatol 0.001 – Step size for differentiation of the log likelihood function for computing the standard errors.
- xblmeth 1 – A value of 1 uses line search as the method for optimizing post-hoc estimates, 2 uses dogleg, 3 uses Levenberg-like trust region.
- xblndig 13 – Estimate of the available accuracy of the optimized post-hoc estimates in terms of decimal places.
- xbltol 0.002 – Step size for differentiation of the model function for FOCE L-B linearization.
- rtol 1E-06 — ODE relative tolerance
- atol 1E-06 — ODE absolute tolerance
- nmxstep 50000 — ODE maximum number of internal allowable steps to achieve the -rtol and -atol accuracies.
- sort – When this argument is included, the input is sorted.
- csv – When this argument is included, the results are saved as a csv file.

The command line to execute the Bootstrap job is in run.sh or run.bat file

Linux (in run.sh):

```
%INSTALLDIR%\bootstrap.bat MULTICORE %INSTALLDIR% %shared_directory%
%workDir% 3 1000 40 3 test.mdl colsl.txt data1.txt 11938 nlmeargs.txt
"nlmeargs.txt colsl.txt data1.txt test.mdl" 4 95 TRUE
```

Windows (in run.bat):

```
${INSTALLDIR}/bootstrap.sh TORQUE_MPI ${INSTALLDIR} ${shared_directory}
`pwd` 3 1000 10 3 test.mdl colsl.txt data1.txt 11938 nlmeargs.txt
"nlmeargs.txt colsl.txt data1.txt test.mdl" 32 95 TRUE
```

Table 4-1. Bootstrap example command options

	Linux	Windows
Parallel Mechanism	MULTICORE	TORQUE_MPI
Installation Directory	%INSTALLDIR%	\${INSTALLDIR}
Shared Directory	%shared_directory%	\${shared_directory}
Local Directory	%workDir%	`pwd`
Method	3	3
Iterations	1000	1000
Number of Samples	10	10
Number of Retries	3	3
Model File	test.mdl	test.mdl
Columns Definition File	cols1.txt	cols1.txt
Data File	data1.txt	data1.txt
Start Seed	11938	11938
Extra Arguments File	nlmeargs.txt	nlmeargs.txt
Files to Copy	"nlmeargs.txt test.mdl nlmeargs.txt cols1.txt data1.txt test.mdl"	"nlmeargs.txt test.mdl nlmeargs.txt cols1.txt data1.txt test.mdl"
Number of Processes	4	32
Confidence Level	95	95

The `run.r` file is the R script that allows the Bootstrap job to be executed from the R program by using the Certara.NLME8 package.

NLME_ROOT_DIRECTORY and PhoenixLicenseServer or NLME_HASH should be set either prior to starting R or from within R session:

R

```
Sys.setenv("NLME_ROOT_DIRECTORY"="/home/nlmeuser/Work")
Sys.setenv("NLME_HASH"="<hash_code_from_Certara>")
setwd("<working_directory>")
source("run.r")
```

Example of bootstrap script:

```
library(Certara.NLME8)
#
# Load Certara NLME library
#
library(Certara.NLME8)
#
# Default setup_env.r creates a Multicore parallel host with 4 cores
#
host=hosts[[1]]
#
# default NLME Dataset consists of Phoenix generated files
# test.mdl cols1.txt data1.txt
#
```

```
dataset = NlmeDataset()
#
# Engine parameters with 1000 iterations and FOCE_LB engine
#
param = NlmeEngineExtraParams(PARAMS_METHOD=METHOD_FOCE_LB,
                              PARAMS_NUM_ITERATIONS=1000)
#
# 50 boot replicates with seed = 1234
#
boot = NlmeBootstrapParams(numReplicates=50,
                           randomNumSeed=1234)
#
# Run a bootstrap job, by default on Linux it will run in the background.
# On windows it waits till the job is finished
#
file.remove("progress.xml")
job=RunBootstrap(host,dataset,param,boot)
#
# Report Progress on the job
#
while (!(NlmeJobStatus(job) == "Finished" || NlmeJobStatus(job) ==
"Failed"))
{
    print(NlmeJobStatus(job))
    print(job)
    Sys.sleep(5)
}
print(job)
```