

PLT Tools

A Graphical Interface for the NONMEM System

User's Manual

Version 6

Dennis M. Fisher MD

Steven L. Shafer MD

www.PLTsoft.com

NOTE: Throughout this manual, files, folders, and operating system commands are displayed in a fixed-width font, Courier. **Buttons** in the graphical interface (GUI) are displayed in a box with a yellow background.

NOTE: All graphics are obtained from the OS X version of the software. The application will appear slightly different in Windows.

UPDATING TO NONMEM VERSIONS 7.41, 7.42, 7.43

In the Workspace/Option tab, **PLT Tools** allows selection of 7.4 but does not distinguish between versions 7.4.0, 7.4.1, 7.4.2, and 7.4.3. If you configured Preferences in **PLT Tools** for an earlier version, then updated to a newer version, **PLT Tools** may continue to point to the older version (you can check the NONMEM outputs to see what version is being run). If **PLT Tools** is not using the correct version, please do the following:

1. Open the **PLT Tools** Support folder (PLTTools-Support). This is easiest to accomplish using the FILE -> OPEN menus in **PLT Tools**. You will see a file named nmfe74.bat (or nmfe74). **DELETE** that file.
2. Return to **PLT Tools**. Open Preferences. In the NONMEM tab, click Select adjacent to NONMEM 7.4. Navigate to the location of the nmfe74.bat (or nmfe74 in OS X), then click the OK button. In Windows, the location is likely to be: C:\nm742\run. In OS X, the location might to be: /opt/nm742/run.
3. In the Workspace/Options tab in **PLT Tools**, select 7.4 as the NONMEM version (you may need to select a different version, then select 7.4).

When you run **PLT Tools**, it should now use the correct NONMEM version. If this does not happen, please contact support@pltsoft.com.

Table of Contents

Introduction	10
Project Controller	10
Graphics Editor	10
Compare/Summarize Runs, Prepare Report	10
Archive Manager	10
Run Listing Editor	11
Control Stream Editor	11
Support for xpose	11
Obtaining and Installing PLT Tools	12
Obtaining PLT Tools	12
Installation	12
NONMEM Versions 5, 6, 7, 7.2, 7.3, 7.4	13
Location of PLT Tools Folders	14
Setting Up PLT Tools	15
Obtaining a License	16
Consultant Mode (available only to licensed users)	17
R	17
Software: nmfe vs. nmqual	18
Location	20
Configuring the Control Stream Editor	21
Parameter Update	22
Multiple Runs (available only to licensed users)	23
Messages	24
Email Notifications (requires configuration)	25
Help	26
Options in the Workspace/Options Window	27
Echo NONMEM (stored)	27
Clear console on run (stored)	27
Remote only (stored)	28

Create XPT Files (stored)	28
NONMEM Version (stored)	28
Add Names (stored)	28
Selecting A Paper Size (stored).....	29
Wider margins (stored)	29
Print Control Stream (stored).....	29
Print objective function (stored)	29
Smoother (stored).....	29
Opening PDF file(s) at completion of NONMEM run (stored).....	29
Compiler option (not stored)	29
Local only (not stored)	29
Suppress non-critical messages (not stored)	30
Seed (not stored):	30
NODES (not stored)	30
Options in the Project Controller Window (Not Stored).....	30
Normal	30
Covariate Search	30
Visual Predictive Check	30
Bootstrap, Number	31
Jackknife.....	31
Likelihood Profile.....	31
Running NONMEM and Creating Graphics	31
Selecting A Workspace in the Workspace / Options Tab.....	31
Text in the Output Window.....	36
Text in the Analysis Window	36
Modifications needed for the NONMEM Control Stream.....	37
Modifications needed if \$PRED is used instead of \$PK.....	39
Modifications to the Dataset	39
Updating Initial Estimates Based on the Output from a Previous Run	39
Adding Records to the Dataset to Smooth PRED/IPRED vs. TIME CURVES.....	40

Entering LOQ Values in the Dataset	43
Conditional Weighted Residuals (CWRES)	43
New Methods in NONMEM	44
NONMEM's CHECKOUT Option.....	45
Providing Covariate Data	45
Starting a New NONMEM Run	46
New Control Stream.....	46
Editing the Control Stream.....	46
Graphics Script	46
Restarting a Terminated NONMEM Run	47
Re-creating Graphics from a Completed NONMEM Run	47
Stopping NONMEM / PLT Tools	48
Simultaneous Runs on a Single Machine or Across a Network.	48
Modes of Operation.....	49
Normal	49
Covariate Search	49
Visual Predictive Check	51
Likelihood Profile.....	51
Bootstrap.....	53
Jackknife Analysis.....	55
Setting a Seed for Bootstrap and Jackknife Analyses.....	56
Skip Base Run (Bootstrap Analysis, Likelihood Profile)	57
Files Created by PLT Tools	58
Brief Summary.....	58
Real-Time Parameter Display	58
Embedded Graphics	59
Preservation of Files Created With Each Run.....	59
Define.TIMESTAMP.csv	61
Rescuing Aborted Bootstraps.....	61
Combining Likelihood Profiles.....	62

Automation	63
Command Line Execution.....	67
Project Log, Parameter Log	68
Project Log	68
Project Log (XL)	69
Parameter Log.....	69
Using the Graphics Editor.....	69
Graphics Script	69
Graphics for each NONMEM run:	69
Data Source for Graphics	72
Recommendations for Source Files.....	74
Truncated IDs	74
Updating Covariates	75
Populating the Graphics Editor	75
Tab 1: THETAS, ETAS, SIGMAS:.....	76
Tab 2: Covariates.....	78
Tab 3: By-Subject Graphics.....	79
Tab 4: Spaghetti Graphics	83
Tab 5: General Layout / Options	87
Tab 6: Text Mapping.....	90
Displaying Greek Characters in Graphics	92
Changes To Graphics Implemented Outside Of Graphics Editor.....	93
Tornado Plots	99
Additional Graphics	101
Data Checkout.....	101
Observations By Subject.....	102
Composite Observations	102
Dose-Normalized Observations.....	102
Covariate Summary	102
Covariate vs. Covariate	102

Visual Predictive Check	102
Standardized Visual Predictive Check	103
Prediction-Corrected Visual Predictive Check.....	103
Between-Subject Grouping Within a VPC	103
Within-Subject Grouping Within a VPC.....	104
Nominal Time vs. Actual Time	104
Likelihood Profile	105
Creating Your Own Graphics	105
Using the Summarize / Report Tool	106
Prepare a Report	106
Assemble Report	109
Elements Accessed from All Runs	109
Elements Accessed from Selected Runs.....	110
Using the Archive Manager Tool	112
Using the Run Listing Editor	115
xpose	117
Obtaining Help	122
Obtaining Help for PLT Tools.....	122
Obtaining Help for NONMEM	123
Assumptions Made by PLT Tools.....	125
Troubleshooting.....	126
Contacting PLTsoft.com	130
Acknowledgements.....	131
Version History.....	132
APPENDIX 1. Setting Up Parallelization of NONMEM Runs.....	149

Figures

Figure 1. The Workspace / Options window.	15
Figure 2. The "Version" tab in the Preferences window.	16
Figure 3. The License tab of the Preferences window.	17
Figure 4. The R tab of the Preferences window. The user must select the path to R.	18
Figure 5. The NONMEM tab of the Preferences window.	20
Figure 6. The Location tab of the Preferences window.	21
Figure 7. Editor tab of the Preferences window.	22
Figure 8. The Parameter Update tab of the Preferences window.	23
Figure 9. The Multiple Runs tab of the Preferences panel.	24
Figure 10. The Messages tab of the Preferences panel.	25
Figure 11. The Help tab of the Preferences panel.	26
Figure 12. The Workspace/Option window is displayed.	27
Figure 13. The Project Folder has been selected.	33
Figure 14. The file structure created by the NONMEM Controller is shown.	34
Figure 15. The Projector Controller tab is displayed.	35
Figure 16. The options fields are configured for a Likelihood Profile.	52
Figure 17. The user can set a seed for bootstrap analyses and certain jackknife analyses by entering a value in the "Seed" field. Only non-negative integers can be entered.	57
Figure 18. An example of a real-time parameter display is shown.	59
Figure 19. A sample automation session is displayed.	64
Figure 20. The pull-down menu displays choices for automation.	67
Figure 21. Tab 1 of the Graphics Editor.	76
Figure 22. Tab 2 of the Graphics Editor.	78
Figure 23. Tab 3 of the Graphics Editor.	80
Figure 24. Tab 4 of the Graphics Editor.	84
Figure 25. The "Header Text" field in the Spaghetti Graphics tab of the Graphic Editor shows two variants of headers that will create "time after dose" graphics.	86
Figure 26. Tab 5 of the Graphics Editor.	88
Figure 27. Tab 6 of the Graphics Editor.	90
Figure 28. A sample tornado plot.	99

Figure 29. The Prepare a Report tool.....	107
Figure 30. Sample Configuration for “Select Runs” in the Summarize / Report Tool.	110
Figure 31. The Archive Manager tool.....	112
Figure 32. The Run Listing Editor is shown.....	115
Figure 33. The lower border of PLT Tools contains two buttons to assist the user with PLT Tools	122
Figure 34. If the user clicks Support , this message appears.	123
Figure 35. If the user clicks NONMEM Help , NONMEM’s help files (provided as part of the NONMEM installation) can be accessed and printed.....	123

Introduction

PLT Tools is a set of graphical tools to facilitate the conduct of NONMEM* analyses on either the Windows or OS X platforms. Use of **PLT Tools** should be instinctive: most features are readily identified in the graphical user interface. Despite this, we provide this manual to guide the user. We welcome suggestions regarding the User's and Installation Manuals as well as potential improvements to the software.

PLT Tools consists of seven elements.

Project Controller

The Project Controller runs NONMEM, creates numerous files containing NONMEM outputs and tables, creates extensive graphics, and performs various statistical analyses from a graphical interface. Folders are created automatically and all files are time-stamped, facilitating Part 11 compliance. A unique feature is a 1-2 page summary ("Brief Summary") showing the objective function, the number of significant digits, final estimates (and standard errors if available) for *thetas*, *omegas*, and *sigmas* and diagnoses of NONMEM boundary conditions.

Graphics Editor

Using the Graphics Editor, the user selects options for the creation of graphics. Although the engine creating the graphics is R, no knowledge of R code is required.

Compare/Summarize Runs, Prepare Report

The Compare / Summarize Runs tool compares one or more pairs of runs, then reports:

- the change in the objective function
- the *P* value for the likelihood ratio test
- line-by-line comparisons of the two Control Streams

The output is a text file that can be inserted as an appendix in a study report.

In addition, a table (csv format; can be embedded on a Word document) summarizes differences between runs, objective functions and the change between models, the *P* value for the likelihood ratio test, the number of non-fixed *thetas*, *omegas*, and *sigmas* in each model, and comments entered by the user.

The Prepare Report tool compiles the output of Compare/Summarize Runs (see above) and the results from one or more NONMEM runs (Control Stream, output) into a rich-text (RTF) document that can be opened by Microsoft Word or Open Office.

Archive Manager

The Archive Manager tool bundles together the outputs of multiple runs into a time-stamped archive. This facilitates transfer or archiving of a completed project.

* NONMEM is a trademark of the Regents of the University of California and ICON.

Run Listing Editor

The Run Listing Editor assists the user in creating a table identifying those runs to be used by the Compare Runs, Prepare Report, and Archive Manager tools.

Control Stream Editor

The user can edit a Control Stream directly from **PLT Tools**, either with the internal editor or an external editor such as Notepad or WordPad (Windows) or TextEdit (OS X). This is accomplished using the “Edit” button in the Project Controller. After editing is complete, the file is saved as a text file with platform-dependent line endings. In addition, the user can access a library of Control Streams that apply to most pharmacokinetic problems (ADVAN 1, 2, 3, 4, 11, 12).

Support for xpose

xpose is an open-source r (r-project.org) package designed to facilitate conduct of NONMEM analyses. **PLT Tools** provides an interface to generate xpose graphics and tables. **PLT Tools** also manages (filing and reformatting, as appropriate) files created for xpose. In addition, **PLT Tools** can create files appropriate for xpose. The user must obtain xpose from the R repository (CRAN); see www.r-project.org.

Obtaining and Installing PLT Tools

Obtaining PLT Tools

PLT Tools can be obtained either *via* download at www.PLTsoft.com or by ordering media from PLTsoft (sales@PLTsoft.com). Versions are available for two platforms, Windows and OS X. Separate distributions, packaged as ZIP archives are available for each platform; if a firewall prevents downloading of a ZIP archive, non-archived versions are also available (contact support@PLTsoft.com). Media containing non-archived distributions for all platforms can be obtained from PLTsoft.com.

Installation

See the Installation Manual for instructions regarding the installation of **PLT Tools**. It is necessary to install R (www.R-project.org) in order to use **PLT Tools**. To obtain all the features of **PLT Tools**, it is necessary to obtain a license. **PLT Tools** can be used with R version 2.9.0 or higher.

Use of **PLT Tools** depends on access to certain files created during NONMEM's installation. NONMEM installation can be performed using installers from either Globomax (ICON) or Metrum.

Globomax (nmfe) installation: If this installer is used, a file nmfe5, nmfe6, nmfe7, nmfe72, nmfe73, or nmfe74 (nmfe5.bat, nmfe6.bat, nmfe7.bat, nmfe72.bat, nmfe73.bat, nmfe74.bat in Windows) is created. Default locations are:

NONMEM Version	Windows	OS X
5	C:\nmv\run\nmfe5.bat	/opt/nmv/run/nmfe5
6	C:\nmvi\run\nmfe6.bat	/opt/nmvi/run/nmfe6
7 (and 7.1, 7.12)	C:\nm7\run\nmfe7.bat	/opt/nm7/run/nmfe7
7.2	C:\nm7\run\nmfe72.bat	/opt/nm72/run/nmfe72
7.3	C:\nm7\run\nmfe73.bat	/opt/nm73/run/nmfe73
7.4	C:\nm7\run\nmfe74.bat	/opt/nm74/run/nmfe74

If the nmfe file is not available, installation will not be successful.

Metrum (nmqual) installation: If this installer is used, a perl file is created – the name of this file is identical to the folder in which the installation is create. For example, if the installation folder is C:\PATH\FOLDERNAME, the file is C:\PATH\FOLDERNAME\Test\FOLDERNAME.pl.

Default locations are:

NONMEM Version	Windows	OS X
6	C:\nmvi\test\nmvi.pl	None suggested
7 (and 7.1, 7.12)	C:\nm7\test\nm7.pl	None suggested
7.2	C:\nm72\autolog.pl	None suggested

However, the user may have selected a different location. Metrum provides configuration files for installation of NONMEM 7 in Windows (using G95) or OS X (using g95 or Intel Fortran).

Default locations for these configurations are:

Platform	Fortran Compiler	Location
Windows	G95	C:\nm7g95\test\nm7g95.pl
OS X	G95	\$HOME/NONMEM/nm7g95/test/nm7g95.pl*
OS X	Intel Fortran	\$HOME/NONMEM/nm7ifort/test/nm7ifort.pl*

* \$HOME is the path to the user's home folder, a folder in /Users; the location of this folder can be identified by opening a Terminal window, then typing:

```
echo $HOME
```

Contact support@PLTsoft.com if you are unable to configure NONMEM, thereby preventing you from installing **PLT Tools**.

NONMEM Versions 6, 7, 7.2, 7.3, 7.4, 7.5

PLT Tools works with NONMEM versions 6, 7, 7.2, 7.3, 7.4 and 7.5. If the user identifies problems with the use of **PLT Tools** with NONMEM 7 or 7.2 (particularly related to new methods), please contact support@PLTsoft.com. Instructions below refer to nmfe; instructions for nmqual are similar.

Adding a new version of NONMEM: If the user adds a new version of NONMEM, the following steps are required in **PLT Tools**:

1. In Preferences, in the NONMEM tab, select the path to the new version.
2. In the Project Controller, select the new version using the "NONMEM" pull-down menu.
3. Select a Working Folder. **PLT Tools** will identify that the appropriate version of nmfe is missing. **PLT Tools** will attempt to locate and copy the file to the PLT Tools Support folder.

Updating from version 7.1 to 7.12: If the user updates from version 7.1 to 7.12, **PLT Tools** will not automatically update -- the nmfe file for both of these versions is named nmfe7.bat (in nmfe7 in OS X) and NONMEM tab in Preferences does not distinguish between these versions. The following steps are required in **PLT Tools**:

1. Access the PLT Tools Support folder from the menu FILE -> OPEN. Delete the file nmfe7.bat (nmfe7 in OS X).
2. In Preferences, in the NONMEM tab, select the path to the new version.
3. In the Project Controller, re-select version 7 using the "NONMEM" pull-down menu.
4. Select a Working Folder. **PLT Tools** will identify that the appropriate version of nmfe is missing. **PLT Tools** will attempt to locate and copy the file to the PLT Tools Support folder.

Reinstalling an existing version of NONMEM: The nmfe file used by NONMEM contains code pointing to the folder in which NONMEM is installed (see the previous section in this manual for default locations). If the user installs a new copy of a previous version of NONMEM, the following steps are required in **PLT Tools**:

1. Access the PLT Tools Support folder from the menu FILE -> OPEN. Delete the file nmfeX.bat (nmfeX in OS X), where X is the version number, e.g., 5, 6, 7, 7.2, 7.3, 7.4 or 7.5.

2. In Preferences, in the NONMEM tab, select the path to the new version.
3. In the Project Controller, re-select the version using the “NONMEM” pull-down menu.
4. Select a Working Folder. **PLT Tools** will identify that the appropriate version of nmfe is missing. **PLT Tools** will attempt to locate and copy the file to the PLT Tools Support folder.

Location of **PLT Tools** Folders

PLT Tools application/executable: During the installation process, the user determines where to install **PLT Tools** (in a folder named PLTTools). The usual location in Windows is either C:\Program Files or C:\Program Files (X86). In OS X, the usual location is /Applications. However, if these locations are not available to the user (*e.g.*, in a locked-down corporate environment), **PLT Tools** can be installed in other locations such as Desktop.

PLTTools-Support: In both Windows and OS X, Preferences and other files relevant to conduct of analyses are stored in a folder names PLTTools-Support. In Windows, the pathway to this folder is determined by an environment variable (APPDATA). In OS X, the folder is located in ~/Library/Application Support. For users unfamiliar with accessing the folder using these descriptions, they folder can be accessed from **PLT Tools**. In the FILE menu, select OPEN, then Support (Preferences) Folder (a keyboard shortcut is provided).

Setting Up PLT Tools

To open **PLT Tools**, click on the application icon. If you have created a shortcut (Windows) or alias (OS X), you can open **PLT Tools** by clicking on that icon. Either approach opens the Workspace / Options window (**Figure 1**).

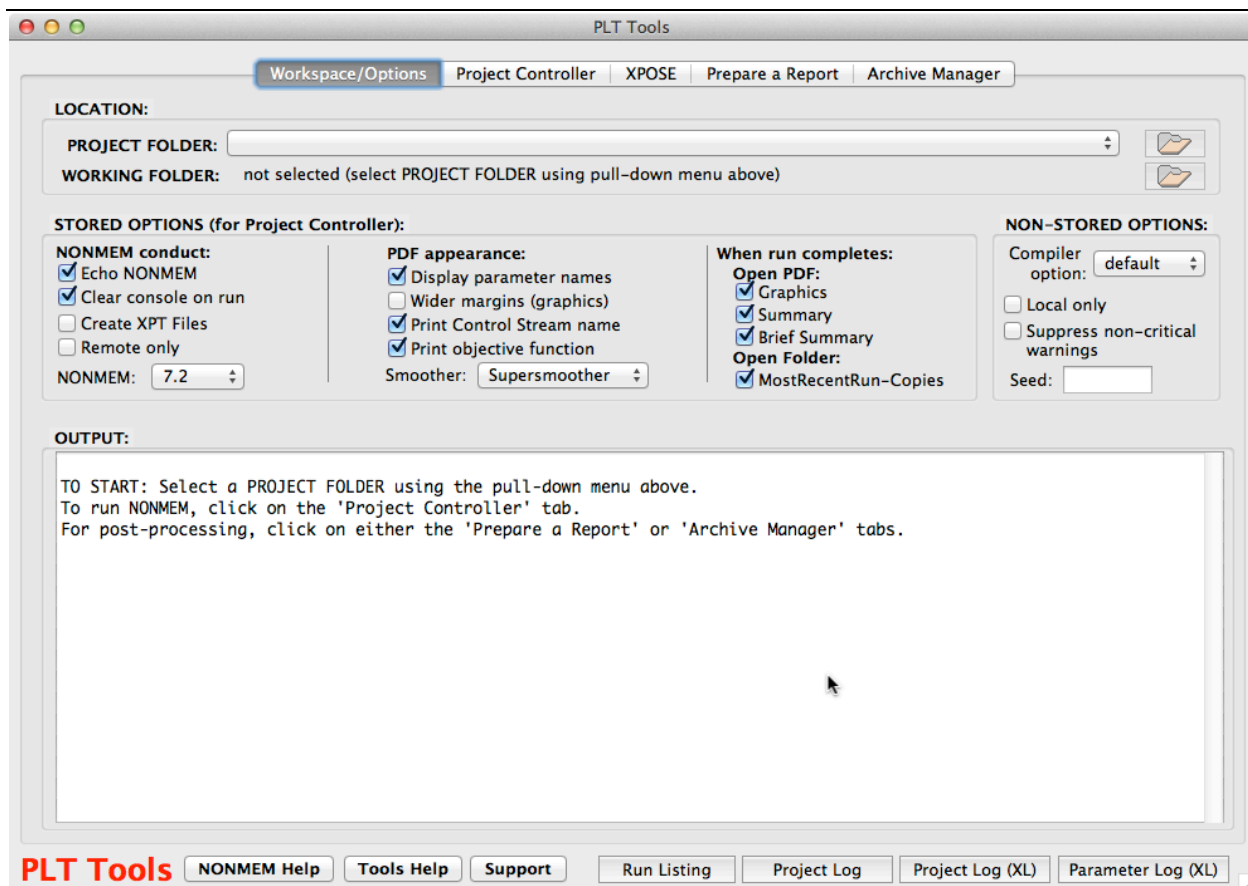


Figure 1. The Workspace / Options window.

In the File Menu, select Preferences. The "Version" tab will appear (**Figure 2**). One option is available in this tab: "Auto-Detect New Versions". If the computer on which **PLT Tools** is installed has internet access and this option is checked, each time that **PLT Tools** is launched, it will check the PLTsoft.com website to determine if a new version is available. If a new version is available, a dialog box will appear, encouraging the user to download the newest version.

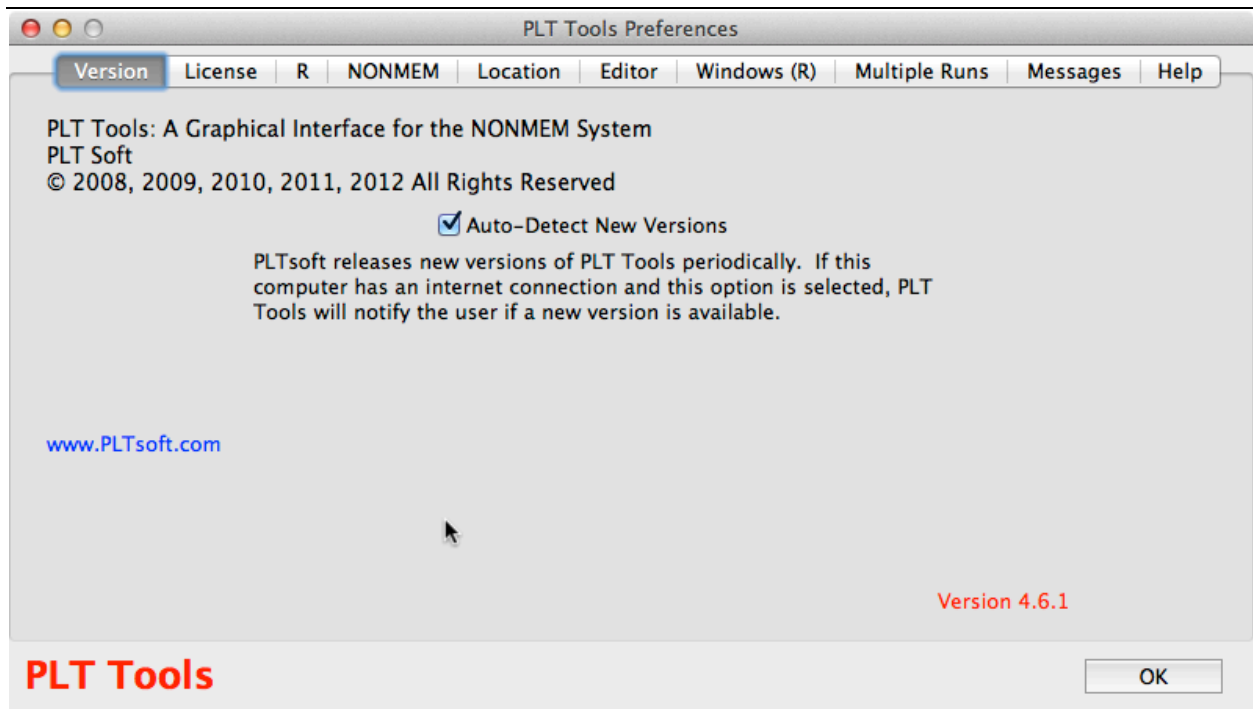


Figure 2. The "Version" tab in the Preferences window.

Obtaining a License

Most features of **PLT Tools** are available without a license. If the user has not obtained a license, the footer of all PDF documents created by **PLT Tools** contains the text:

Created with an unregistered version of PLT Tools (register at PLTsoft.com)
If the user obtains a license, the footer of PDF documents created by **PLT Tools** contains the text:

Property of COMPANYNAME (Confidential)

where the user selects COMPANYNAME.

To obtain all features, including support, the user needs to obtain a license. Details on obtaining a license, including a purchase form, can be found at PLTsoft.com. Before submitting a license request to PLTsoft, the user should install **PLT Tools** to confirm that installation can be accomplished.

Once a license code is obtained, open Preferences, then go to the License / Registration tab (**Figure 3**). Enter the Company / Institution name exactly as it was transmitted to PLTsoft. Then, enter the Registration Code that you received from PLTsoft. Next, click **Register**. The text "Unregistered" should be replaced with an indication that you have registered successfully and showing the expiration date of the license. Contact support@plessthan.com if this procedure is not successful.

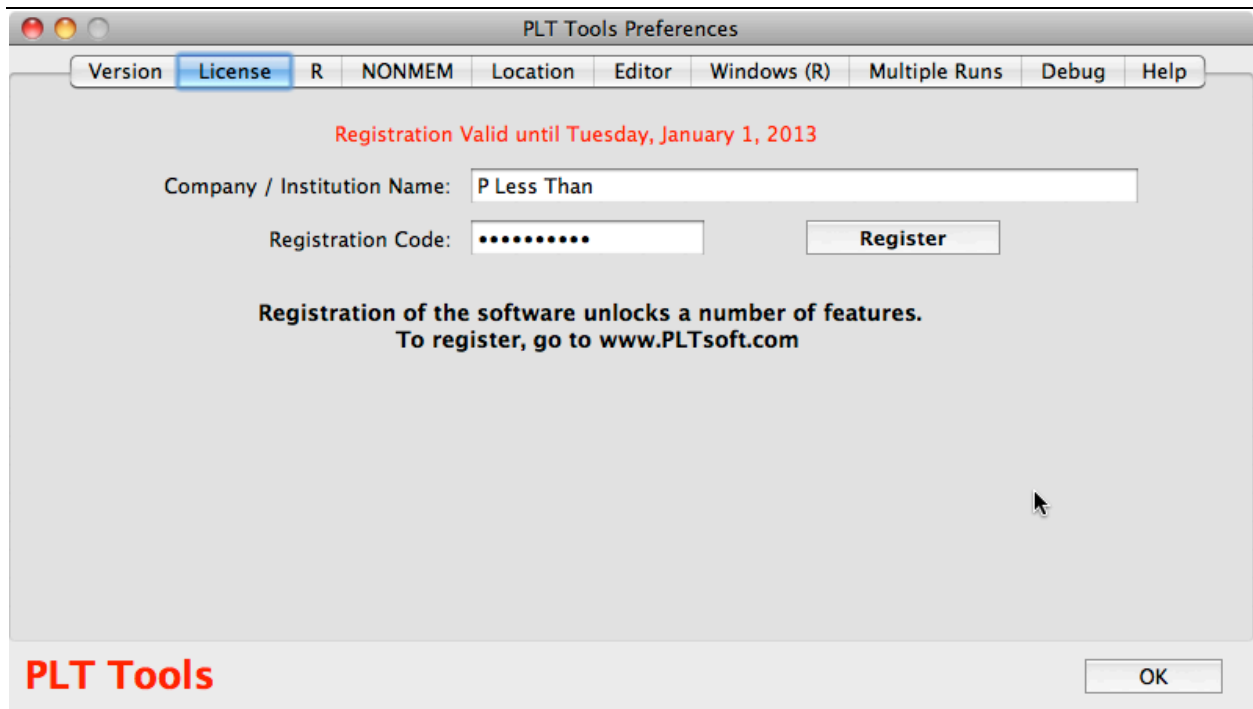


Figure 3. The License tab of the Preferences window.

Consultant Mode (available only to licensed users)

A registered user may wish to change the `COMPANYNAME` that appears in the footer (*e.g.*, a consultant may wish for his/her client's name to appear). This can be accomplished using the following steps:

1. In the `ProjectFolder`, create a file named `REGISTRATIONOVERRIDE.txt` (case-sensitive).
2. That file should contain the text that will replace `COMPANYNAME`.

If **PLT Tools** detects the presence of `REGISTRATIONOVERRIDE.txt` and the user is registered, the content of that file will replace `COMPANYNAME` in the footer. In addition, the text

PLT Tools licensed to `COMPANYNAME`

will appear at the right margin of all PDF documents.

R

PLT Tools uses R as its internal engine.* Open Preferences, then, navigate to the Software tab (Figure 4). Click on the **SELECT** button in "Path to R", navigate to the appropriate location then, enter SELECT.

* Support for S-Plus in **PLT Tools** was discontinued at version 2.5 of **PLT Tools**. S-Plus did not permit a number of important features of **PLT Tools**.

Usual locations for R are (**note the selection of Rterm in Windows**):

Windows: \Program Files\R\R\3.4.1\bin\Rterm.exe

OS X: /Library/Frameworks/R.framework/Versions/3.4/Resources/bin/R

If you have installed a newer version of R, change the text “3.4.1” or “3.4.2” accordingly.

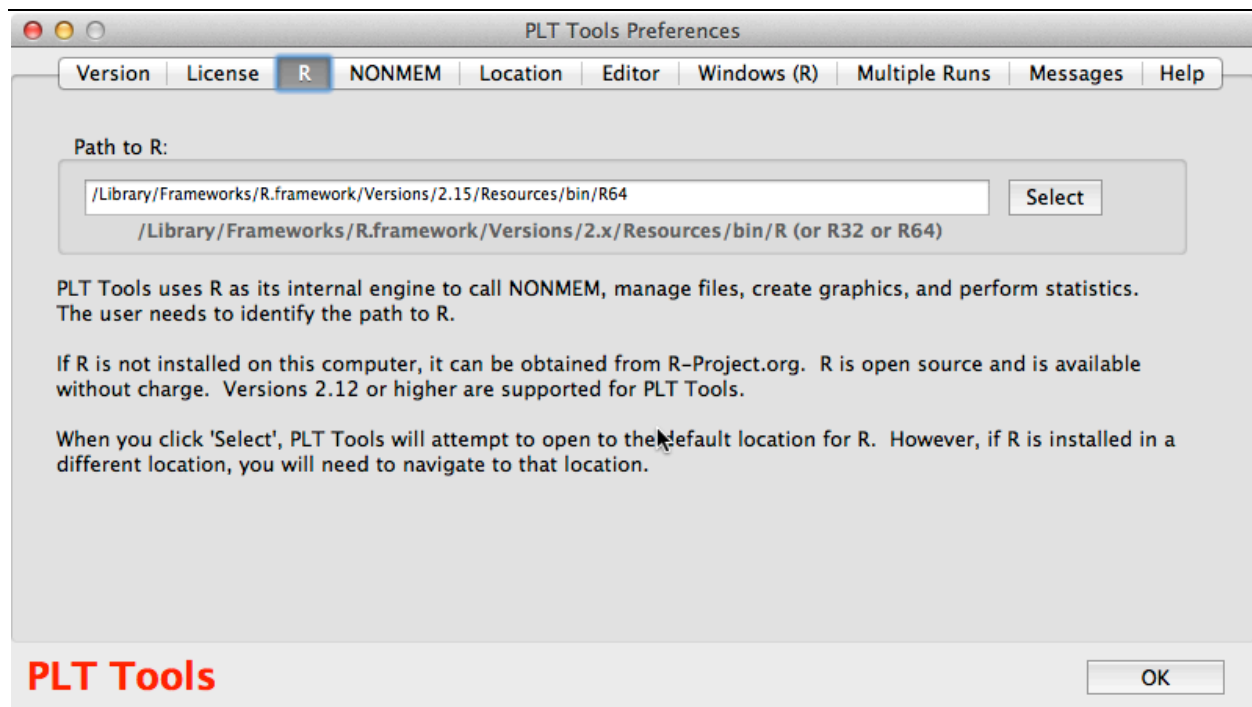


Figure 4. The R tab of the Preferences window. The user must select the path to R.

Software: nmfe vs. nmqual

PLT Tools allows selection of either nmfe (Globomax / ICON) or nmqual (Metrum) as the engine to run NONMEM (**Figure 5**). If both are installed, the user can switch simply by selecting the desired engine.

Two steps are required:

1. Select between nmfe and nmqual.
2. Browse to the location of the NONMEM installation. Search the folder identified below for the appropriate file:

nmfe:

Default NONMEM location: see Installation (above)

Folder: run

File:

Windows:

NONMEM 5: nmfe5.bat

NONMEM 6: nmfe6.bat

NONMEM 7: nmfe7.bat
NONMEM 7.2: nmfe72.bat

NONMEM 7.3: nmfe73.bat
NONMEM 7.4: nmfe74.bat
NONMEM 7.5: nmfe75.bat

OS X:

NONMEM 5: nmfe5
NONMEM 6: nmfe6
NONMEM 7: nmfe7
NONMEM 7.2: nmfe72
NONMEM 7.3: nmfe73
NONMEM 7.4: nmfe74
NONMEM 7.5: nmfe75

nmqual:

Default NONMEM location: see Installation (above)

Folder: test

File: extension is .pl (perl). Filename is identical to foldername, *e.g.*,
C:\nm7-nmqual\test\nm7-nmqual.pl

If you use Intel Fortran in Windows, you need to perform one additional step during the installation process. See the yellow-highlighted text in Troubleshooting in this manual.

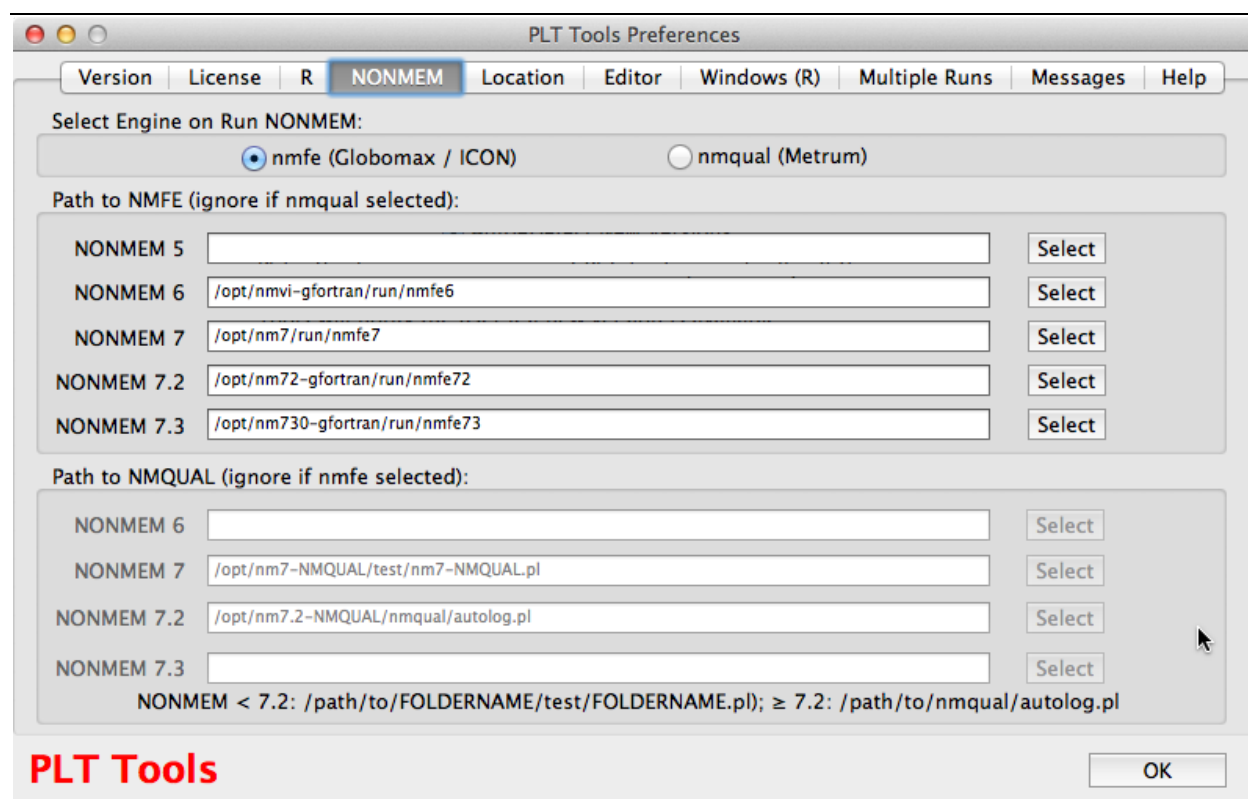


Figure 5. The NONMEM tab of the Preferences window.

After selecting nmfe or nmqual as the engine to run NONMEM, the path to the NONMEM installation must be selected.

Location

PLT Tools reformats NONMEM output tables to allow them to open in Excel or OpenOffice (source files are also archived). These new tables, in addition to other tables created by **PLT Tools** can be saved in different formats, selected by the user. Delimiter options are:

- comma (typical in North America, .csv extension)
- semicolon (typical outside North America, .csv extension), and
- tab (.tab extension).

PLT Tools also creates PDF documents – the size of these documents depends on the setting selected here. Choices are:

- 8.5 x 11: typical in North America
- A4: typical outside North America.

The user configures these options in the “Location” tab in Preferences (**Figure 6**). During the initial installation of **PLT Tools**, an attempt is made to set these preferences based on the local time zone (North American time zones result in the selection of comma and 8.5 • 11 paper; other time zones result in the selection of semicolon and A4 paper).

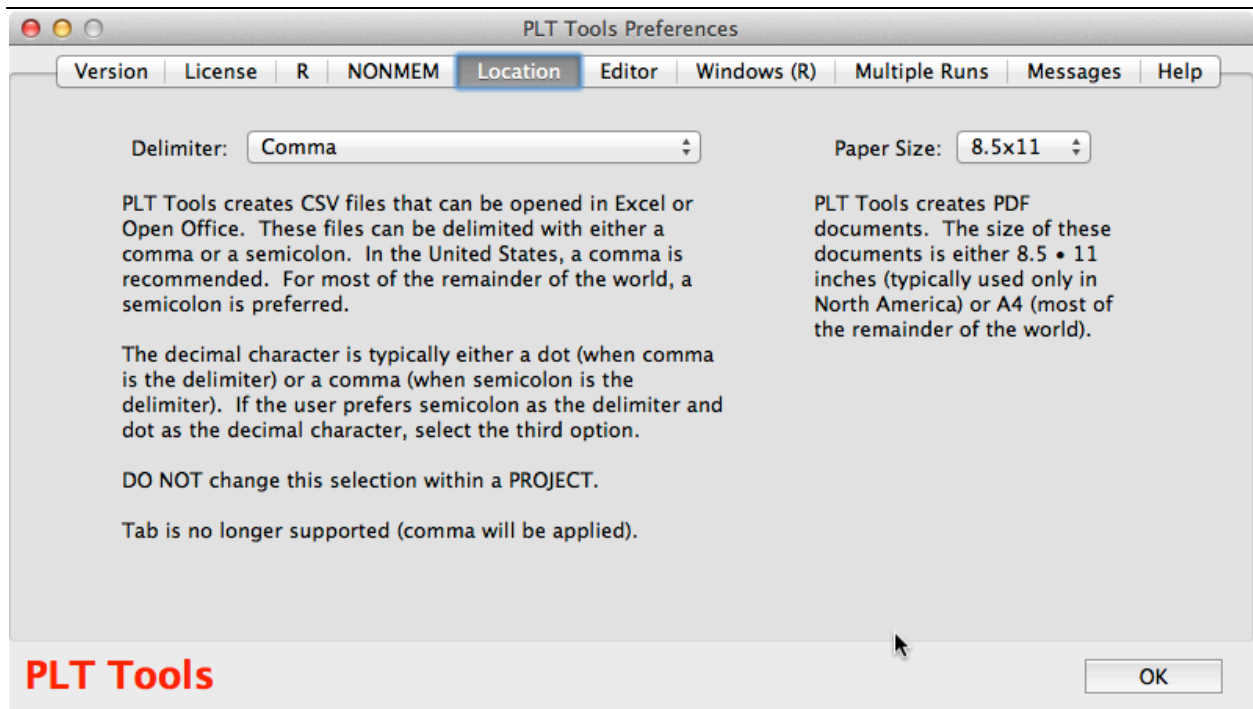


Figure 6. The Location tab of the Preferences window.

The user must select the delimiter for tables created by **PLT Tools** and paper size for PDF files created by **PLT Tools**.

Configuring the Control Stream Editor

The Project Controller tab of **PLT Tools** contains an **Edit** button for Control Streams. If the Editor tab in Preferences (**Figure 7**) is not configured, the internal editor provided by **PLT Tools** is used. The user can control font size for that editor using a pull-down menu in Preferences -> Editor. In addition, the user can configure **PLT Tools** to access an external editor. There are two means to accomplish this:

1. If the “path” to the editor is known to your operating system (e.g., if typing EDITORNAME [such as Notepad or TextEdit] in a Command Prompt or terminal window opens the editor), enter EDITORNAME in the “Editor Path” field.
2. If the “path” to the editor is not known to your operating system, use the **Select** button to browse to the executable or application.

If the latter approach is used, quotes may appear around the path – this may be necessary to accommodate spaces in the path.

If the user wishes to invoke options for the editor, enter those to the right of the path (outside the quotes, if quotes appear). Contact support@plessthan.com if this procedure is not successful.

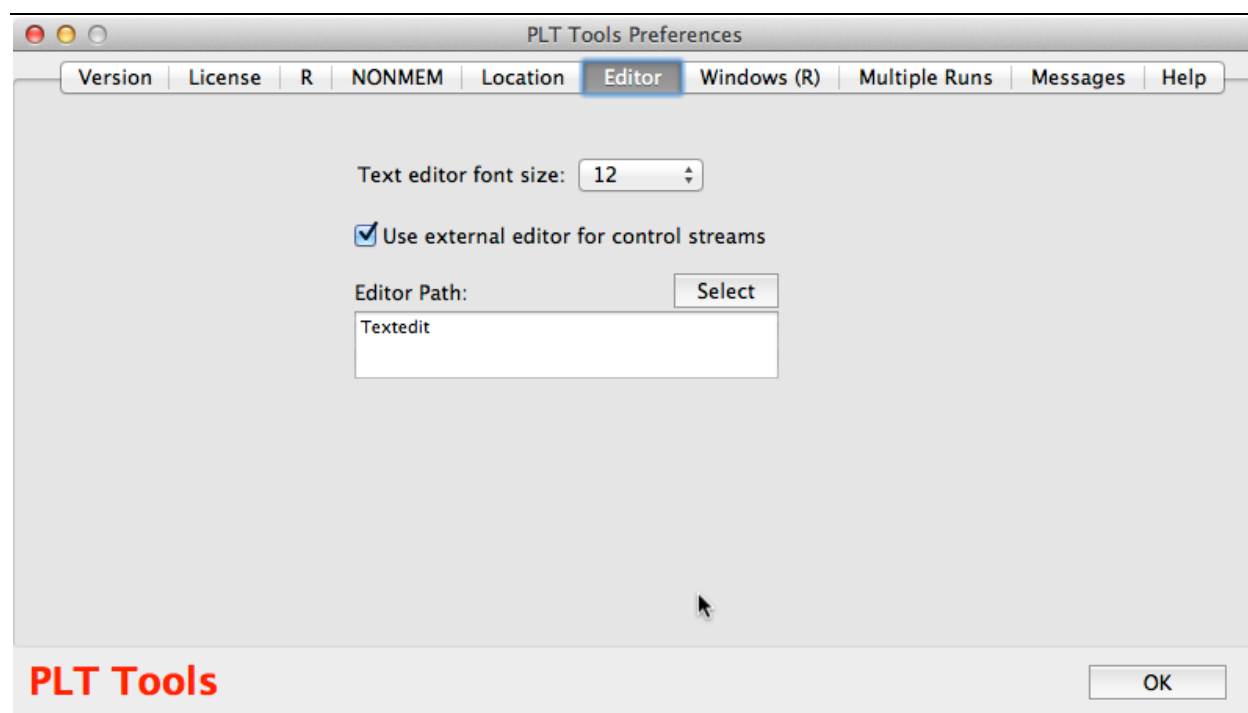


Figure 7. Editor tab of the Preferences window.

Parameter Update

During NONMEM runs, a text file INTER provides regular updates on the values of parameters. **PLT Tools** displays values for the objective function and THETAs immediately after the file is updated. The graphic display enables the user to track the progress of the NONMEM analysis, both the rate of change of the objective function and the course of the parameter estimates.

Creation of the INTER file by NONMEM requires that the \$EST record include a statement "MSF=msfo.outputfile" or "MSFO=msfo.outputfile". If the user omits this text (and has not unchecked "Display live parameters during run", the Control Stream is modified to include the required text.

The user can prevent the display from appearing by unchecking "Display live parameters during run" in the Parameter Update tab of Preferences (**Figure 8**). The size of the window can be altered.

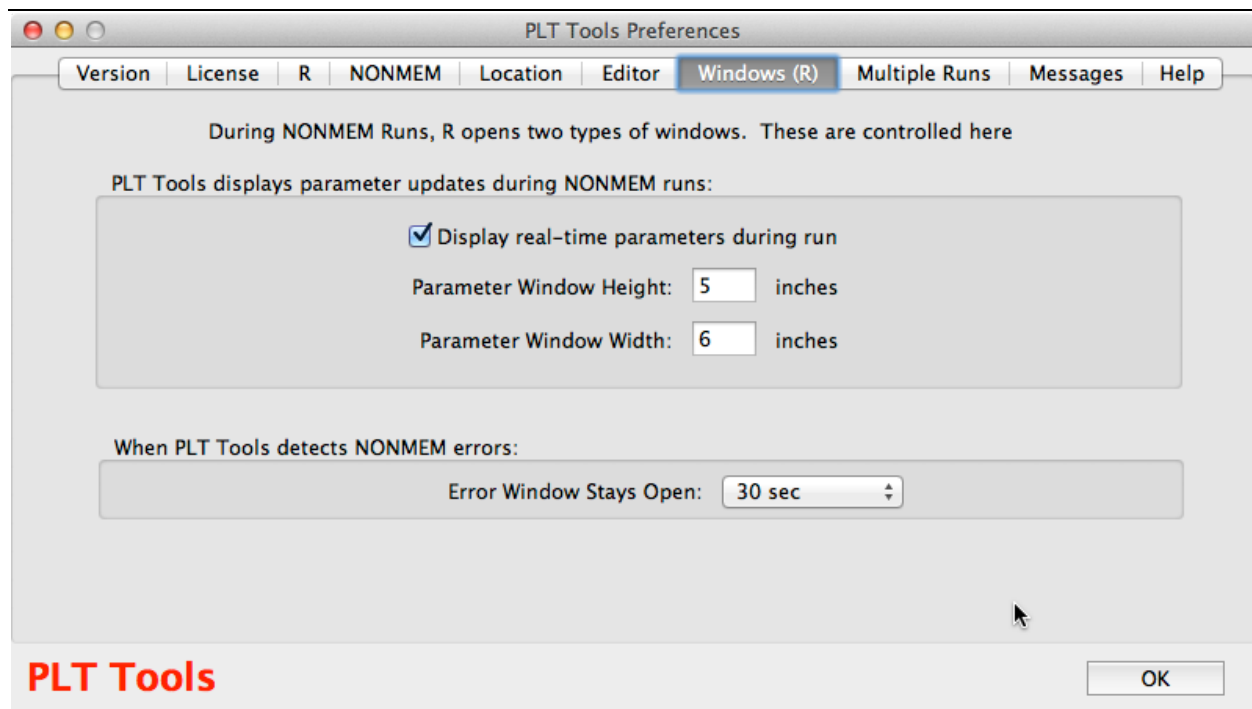


Figure 8. The Parameter Update tab of the Preferences window.

Multiple Runs (available only to licensed users)

Licensed users can perform multiple simultaneous NONMEM runs on a single computer. This is advantageous if a multi-core (multi-processor) computer is being used. During “Normal” and VPC NONMEM runs, each instance (window) of **PLT Tools** generates a single NONMEM run whereas likelihood profiling and bootstrap and jackknife analyses generate multiple runs. Licensed users can perform these multiple runs in parallel, using all processors and speeding throughput.

Multiple simultaneous runs (“parallel runs”) are controlled from this panel (**Figure 9**). Checking “Allow Parallel Runs” activates the feature.

During parallel runs, the user can control the number of simultaneous runs. The default value is the number of processors (determined by **PLT Tools**) minus 1. This keeps one processor in reserve so that the user can perform other NONMEM runs. However, the user can perform other NONMEM analyses.* For example, if a quad-core computer is in use, three processors can be used for a bootstrap analysis, the remaining processor reserved for another analysis.

* For example, if a user initiates a bootstrap analysis that will take several days, it may be necessary to perform other NONMEM runs during that period. Reserving one or more processors will allow the user to conduct both the bootstrap analysis and other analyses.

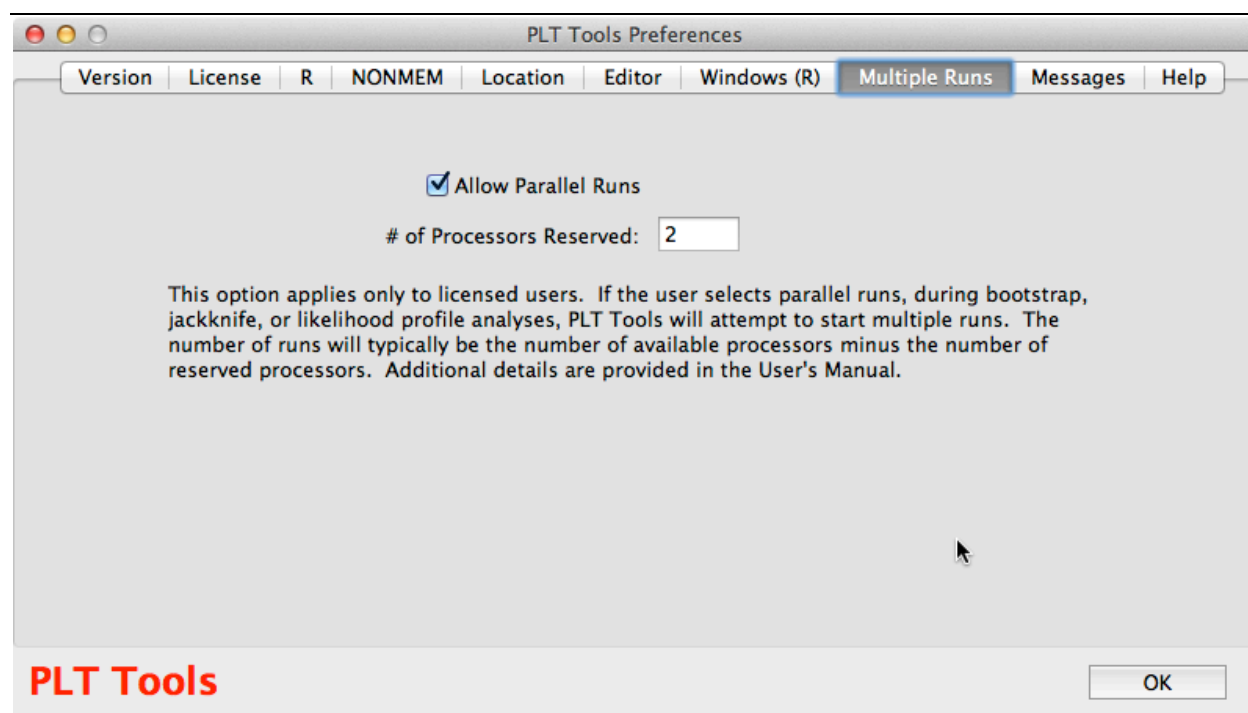


Figure 9. The Multiple Runs tab of the Preferences panel.

Messages

PLT Tools sends many messages to the console. Once a user is familiar with these messages, they may no longer be useful. The user can control certain of these messages.

Include Debugging Output: If this option is selected (**Figure 10**), **PLT Tools** outputs additional debugging information to the Output window. Typically, this option should NOT be checked.

Do not display "Boundary Condition" message in pop-up: NONMEM often reports 'boundary conditions'. Checking this option prevents a message from appearing in a pop-up window.

Add \$MSFO=MSFO.OUTPUTFILE to \$ESTIMATION in Control Stream without notifying user: **PLT Tools** uses a file named INTER to create graphics showing real-time updates of parameters. The INTER file is created only if the user specifies that NONMEM should create a model specification file. If the user omits this step, **PLT Tools** adds the instruction to the Control Stream. Checking this option prevents a message from appearing in a pop-up window.

Do not notify user about limitations associated with \$PRED: If the Control Stream contains \$PRED rather than \$PK, AMT, EVID, RATE, and CMT (and possibly other dose-related data items) will be ignored. Checking this box suppresses an explanatory message.

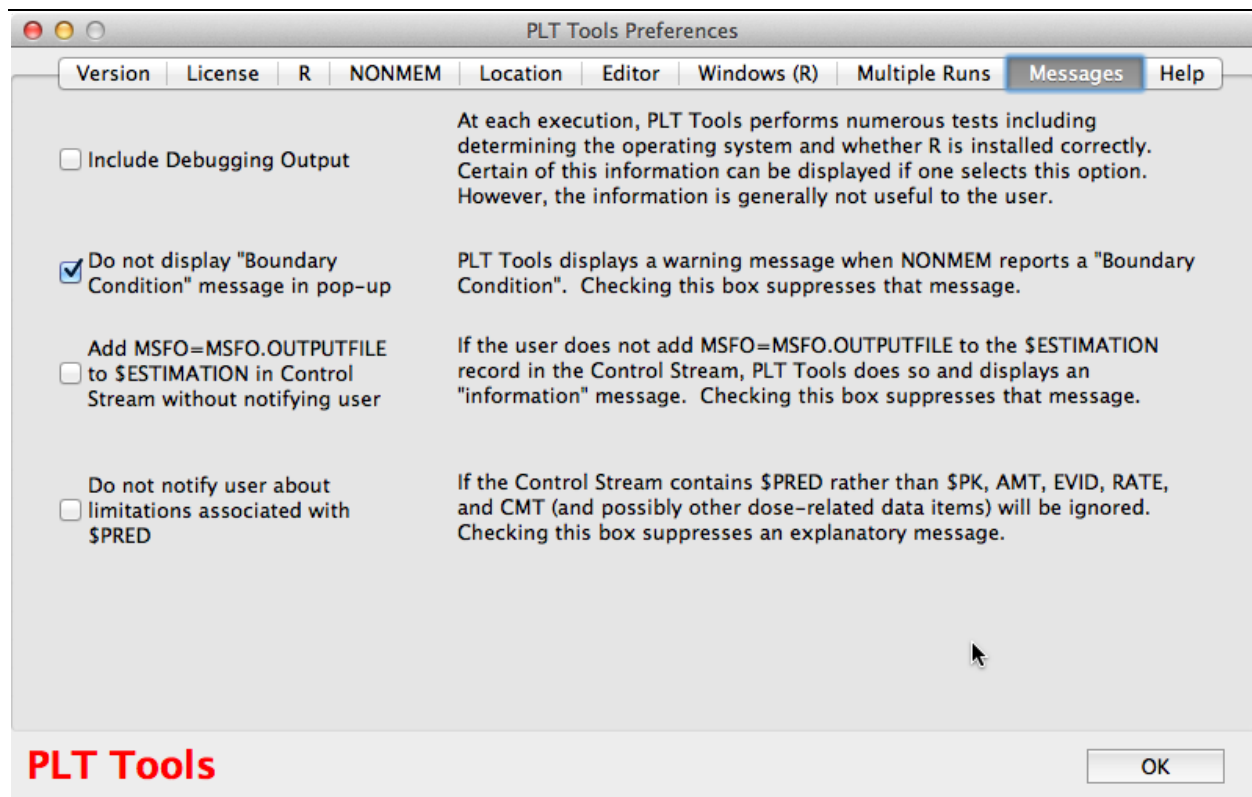


Figure 10. The Messages tab of the Preferences panel.

Email Notifications (requires configuration)

PLT Tools can send email notification to the user under the following circumstances:

- a. A normal run is complete
- b. The Nth iteration of a likelihood profile or a bootstrap analysis is complete.
- c. A likelihood profile or bootstrap analysis has finished.

Emails are sent from a PLTTools.Notify@gmail.com (you may need to inform your email client program that mail from that address is not spam) to an email address that the user designates.

The email notification informs the user of the status of the run and (at completion of a normal run) includes the Brief Summary file.

Configuration is required:

1. The user must provide an email address in the Email tab in Preferences
2. The user must install the R package `emayili`. This can be accomplished by opening R (or Rgui or RStudio) and typing:


```
install.packages("emayili", quietly=T, repos="cloud.r-project.org")
```

 This needs to be done only once. The installation can also be accomplished using the "Install Packages" menu items in RStudio or the Package Installer in Rgui; please select the `cloud.r-project.org` repository.

Once configuration is complete, **PLT Tools** will send email when the following text is identified in the Control Stream:

```
; PLTTOOLS: SENDMAIL
```

or

```
; PLTTOOLS: SENDMAIL NN
```

where NN is an integer. **PLT Tools** will attempt to send email:

- after any "normal" NONMEM run with that Control Stream
- after NN iterations of a bootstrap analysis or a likelihood profile
(if you do not enter an integer for NN, the default value (10) will apply)
- when a bootstrap analysis or a likelihood profile completes.

If you are unsuccessful at sending emails, please contact support@PLTsoft.com.

Help

PLT Tools provides access to NONMEM help files provided with the NONMEM installation. During the initial setup, **PLT Tools** attempts to find the help files and auto-populate this Preference (**Figure 11**). If this procedure is not successful, the user will need to populate the field. Default locations are displayed. In OS X, /opt does not normally appear in the Finder. To access /opt (e.g., if NONMEM is located in /opt/nm7), type "/opt" in the search field in a Finder window.

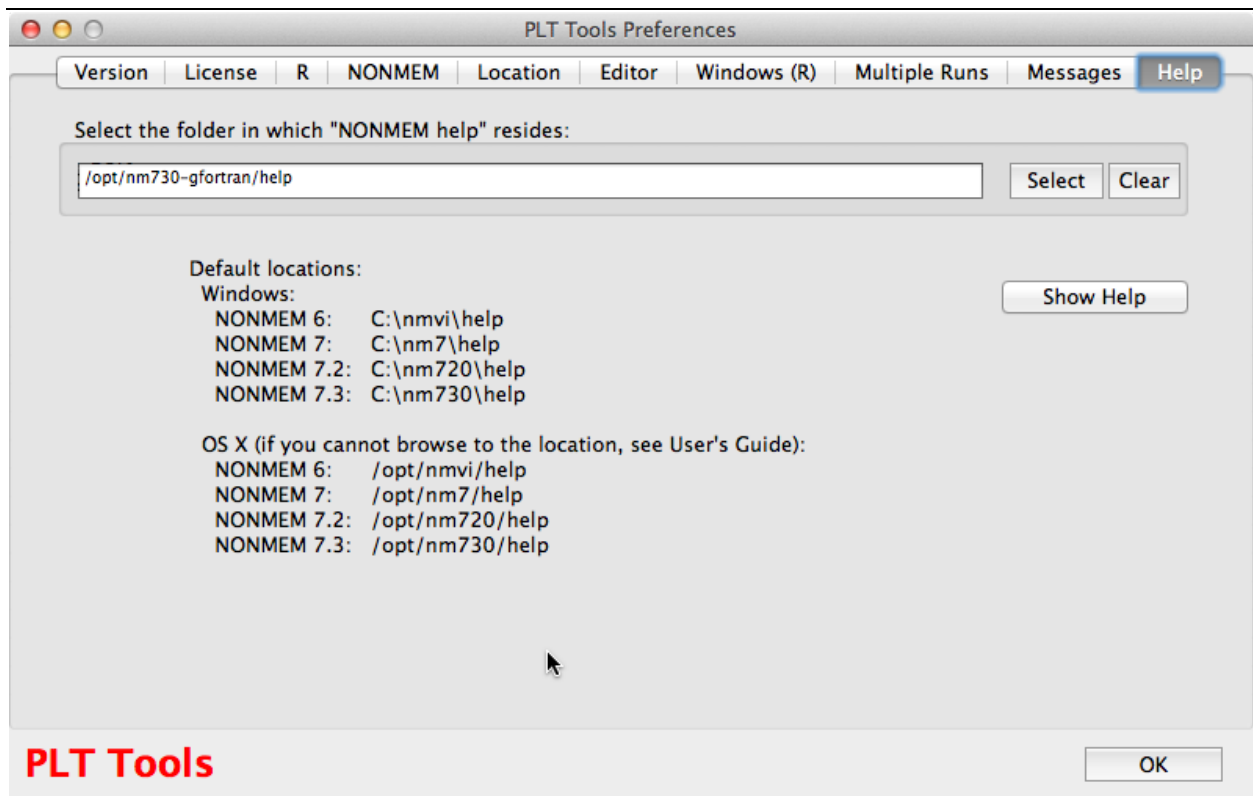


Figure 11. The Help tab of the Preferences panel.

Options in the Workspace/Options Window

A variety of options can be selected in the Workspace/Options window (Figure 12).

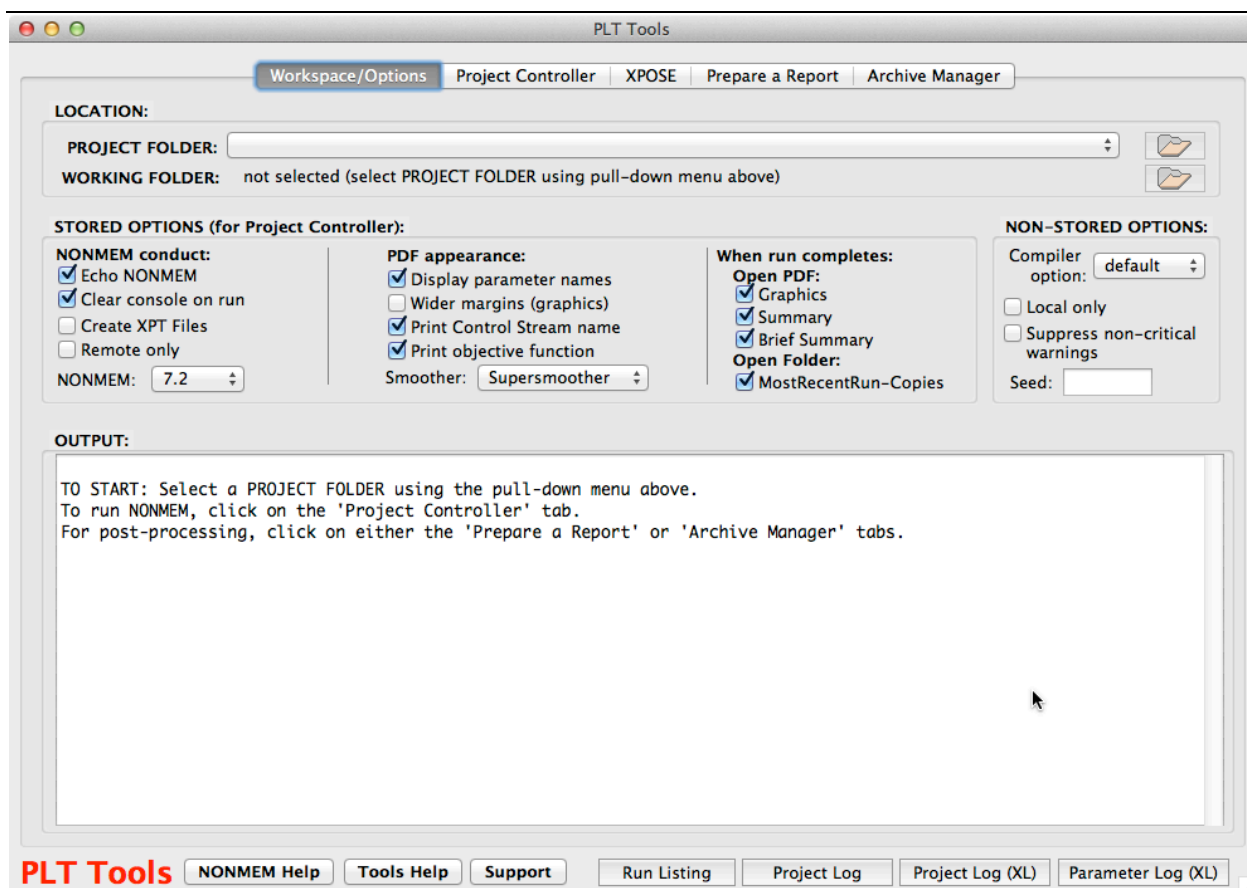


Figure 12. The Workspace/Option window is displayed.

Echo NONMEM (stored)

Certain Fortran compilers do not send their “standard output” to the Output window while it is being written to the NONMEM outputfile. This limits the user’s ability to gauge the status of a NONMEM run, particularly if that run is lengthy or if it is prone to aborting. If the “Echo NONMEM” option is selected, **PLT Tools** captures the contents of the outputfile in real-time and copies them to the Output window. If the Fortran compiler does send its output to the Output window and this option has been selected, redundant text will appear in the Output window. This can be remedied by de-selecting the option.

Clear console on run (stored)

The Output window in Project Controller receives output indicating the status of **PLT Tools** and of the NONMEM run. The user can clear the contents of the window manually by clicking the “Clear” button above the OUTPUT window. If the user selects this option, the console is cleared automatically at the beginning of each run.

Remote only (stored)

In certain situations, the user does not have NONMEM on the computer on which **PLT Tools** is installed; in these instances, NONMEM must be run on a remote machine *via* a network. If the user selects this option, setup procedures searching for a local installation of NONMEM are suspended. Network access requires registration.

Create XPT Files (stored)

Regulatory agencies may request that data files be submitted in SAS XPORT format. To accommodate this request, **PLT Tools** can reformat the `FDATA` file created by NMTRAN (NMTRAN reformats the input data file to a format that can be read by NONMEM). If this option is selected, **PLT Tools** reformats the `FDATA` file for each run and saves it in SAS XPORT format with the extension `.xpt`. In addition, the column names associated with the `FDATA` file (obtained from the `$INPUT` statement in the Control Stream) are recorded in a text file "ColumnNames".

This procedure may be slow – if it exceeds 15 seconds, a message is sent to the Output window. The user can disable this feature if the time is excessive and XPT-formatted files are not needed.

Note that this option creates a separate XPT file for each run and this file is based on the file created by NMTRAN rather than the input file. If the "drop" or "skip" options are invoked in the `$INPUT` record in the Control Stream, `FDATA` differs from the input file.

PLT Tools can also convert input files to SAS XPORT format – see instructions for the Archive Manager.

The conversion to XPT format uses an R package named `haven`. If `haven` is not installed on your computer (or if you installed an earlier version), **PLT Tools** will prompt you to install the appropriate version. For those not accustomed to installation of packages in R, the process is most readily accomplished using any of the graphics interfaces for R (*e.g.*, R Studio or Rgui): search the menus for "Install Packages", then select `haven`. If the option "Install Dependencies" is available, select that option.

NONMEM Version (stored)

NONMEM version 5, 6, 7, 7.2, 7.3, and 7.4 are all supported by **PLT Tools**. The user selects the version using this pull-down menu.[†]

Add Names (stored)

If this option is selected, **PLT Tools** examines the Graphics Script (if one has been selected) to determine if `THETA#` values were entered via the Graphics Editor. If so, these names, when available, appear in the left column of the Brief Summary document. In addition, if `THETA#`

[†] In versions before 2.3.0, NONMEM 5 could be selected by creating a file `NONMEM5.txt` or by including the text `NONMEM5` in the Control Stream. This option is no longer available.

values were entered, the names corresponding to OMEGA values are displayed in the left column of the Brief Summary document. If this option is not selected, the THETA and OMEGA sections of the Brief Summary documents do not contain parameter names.

Selecting A Paper Size (stored)

PLT Tools allows selection of two paper sizes: 8.5 • 11 inches or A4.

Wider margins (stored)

In order to optimize use of the printed page, **PLT Tools** normally allows a margin of ½" at each side of a printed document. Selecting "Wider Margins" increases these margins to 1". These wider margins may be necessary for regulatory submissions.

Print Control Stream (stored)

If the user selects this option, the path to and name of the Control Stream appears in the lower left corner of each page of graphics.

Print objective function (stored)

If the user selects this option, the minimum value of the objective function (if available), as reported by NONMEM, appears in the lower left corner of each page of graphics (with a label indicating that the value is the objective function).

Smoother (stored)

PLT Tools adds smoothers to certain graphics. The user can select between Supersmoother, lowess, and loess.

Opening PDF file(s) at completion of NONMEM run (stored)

PLT Tools is set to automatically open PDF files at the completion of a NONMEM run. To disable this option, uncheck the appropriate boxes in the Settings tab of the Preferences window. If the PDF files do not open despite having allowed the option, there may be a problem with the configuration of your operating system. The most common problem is that you have not installed a PDF viewer, e.g., Acrobat (Windows, OS X) or Safari (OS X). A second possibility is that the PDF viewer is not configured as the default application to open PDF files. If you are unable to resolve this problem, contact support@PLTsoft.com.

Compiler option (not stored)

NONMEM 7.2 allows the user to select different options for real-time Fortran compilation – see instructions provided with NONMEM 7.2. This option is ignored for older NONMEM versions.

Local only (not stored)

This option is relevant only to users with networked systems. If a computer is removed from the network (e.g., taking home a laptop computer), **PLT Tools** will attempt to access the

network, potentially delaying startup. Selecting this option forces **PLT Tools** to recognize that the network is not available.

Suppress non-critical messages (not stored)

PLT Tools send various messages (information, warnings, errors) to the OUTPUT window. Certain of these messages also appear in a pop-up window that appears at completion of a run. If the user considers the information messages as non-informative, select this option. The next time that an analysis is performed in that WORKINGFOLDER, the pop-up box will include a special message indicating that informational messages will not appear for that folder until **PLT Tools** is restarted. Warnings and errors will not be altered by this option.

Seed (not stored):

When **PLT Tools** performs Bootstrap and Jackknife analyses, subsets of data are selected for each run. In order to select these data subsets, **PLT Tools** uses a random number generator. The user can select the “seed” (starting point) for the random number generator by entering a value here. If a value is not selected and the user performs either Bootstrap or Jackknife analyses, **PLT Tools** assigns a value for seed. The selected value is recorded in NONMEM output. Access to this seed value allows a user to replicate Bootstrap and/or Jackknife analyses if necessary.

NODES (not stored)

If the user has configured NONMEM for parallel runs, **PLT Tools** can execute parallel runs. Select the number of nodes that you wish to run. In addition to configuring NONMEM for parallel runs, one-time steps are required in **PLT Tools**. These steps are detailed in **Appendix 1**.

Options in the Project Controller Window (Not Stored)

Normal

This is the default mode in which NONMEM is conducted without a covariate search, visual predictive check, bootstrap, likelihood profile, or jackknife procedure.

Covariate Search

Checking this box informs **PLT Tools** that a Covariate Search should be performed. See Covariate Search.

Visual Predictive Check

Checking this box informs **PLT Tools** to perform a Visual Predictive Check. See Visual Predictive Check

Bootstrap, Number

Checking the Bootstrap box informs **PLT Tools** to perform a bootstrap. See Bootstrap.

Jackknife

Checking this box informs **PLT Tools** to perform a jackknife analysis. See Jackknife Analysis.

Likelihood Profile

Checking this box informs **PLT Tools** that a likelihood profile should be performed. See Likelihood Profile.

Running NONMEM and Creating Graphics

Selecting A Workspace in the Workspace / Options Tab

PLT Tools requires that the name of the folder in which the Control Stream is located by named in a specific manner – it must contain the word “work” within its name (case-insensitive). This is necessary because **PLT Tools** creates many folders; the rule ensures that **PLT Tools** does create not these folders in undesirable locations. Folders required by **PLT Tools** are:

1. **Project Folder**: This folder can be located at any level on any hard drive. There is only one restriction imposed by **PLT Tools** on the naming of the Project Folder: the path to the folder and the folder itself cannot contain special characters (*e.g.*, [,], {, }, /, \, *).
2. **Working Folder**: The Working Folder, located within the Project Folder must contain the word “work” within its name (case-insensitive). For example, acceptable names are Working Folder, Work Folder, Approach1–Working, Working–Approach1.

The first step in using **PLT Tools** is to create these folders. There are two means to accomplish this, one using **PLT Tools**, the other using other applications on your computer.

Using **PLT Tools**: In the FILE menu, select “New Project”. A menu will appear – select the location of the folder. There are no restrictions on naming the folder (however, metacharacters [*e.g.*, [,], /, \] or spaces may cause problems with the operating system. **PLT Tools** will create that folder and the necessary sub-folders.

Using the operating system: Using your usual tools (*e.g.*, Command Prompt or Explorer in Windows or a terminal window in OS X), create a Project Folder; choose any name allowed by the operating system (except that the path to the folder cannot contain special characters). Within that folder, create a folder named WORK (the name must contain the letters “work” in sequence, case-insensitive; longer names are acceptable).

The user must create these folders *before* selecting them in **PLT Tools**. Once the folder are created, in the Workspace / Options tab, click on the **Select** button. Navigate to the Working Folder. Then press “Select”. Once **PLT Tools** confirms that the Working Folder is named appropriately (*i.e.*, it contains “work”, case-insensitive), the file structure shown in **Figure 14** is

created (the user can place input files in any location allowed by NONMEM; the DATAFILES folder is provided for convenience). If the Working Folder selected by the user does not contain “work”, folders are not created; this assures that folders are created only within a Project Folder.

Two folders, Working Folder (the folder selected by the user) and Project Folder (one level higher in the file structure) are displayed. The Project Controller executes its tasks in the Working Folder; Compare Runs and Create Archives execute their tasks in the Project Folder.

For each NONMEM project, the following steps should be followed:

1. Using tools from the operating system, create a folder (referred to as the Project Folder) anywhere in the file system. There is only one restriction imposed by **PLT Tools** on the naming of the Project Folder: the path to the folder and the folder itself cannot contain special characters (e.g., [,] , { , } , / , \ , *).
2. Within the Project Folder, create a Working Folder. The Working Folder must contain the word “work” within its name (case-insensitive). For example, acceptable names are WorkingFolder, WorkFolder, Approach1-Working, Working-Approach1.
3. In the Workspace / Options tab in **PLT Tools**, use the pull-down menu to select the Project Folder. Once a Project Folder has been selected, it is added to the menu to facilitate future selections (the menu can be cleared or edited by the user). If the Working Folder (and its enclosing Project Folder) is found (**Figure 13**), **PLT Tools** will display its full path; in addition, all necessary folders will be created (**Figure 14**). If the Working Folder is not found, an error message will be displayed.

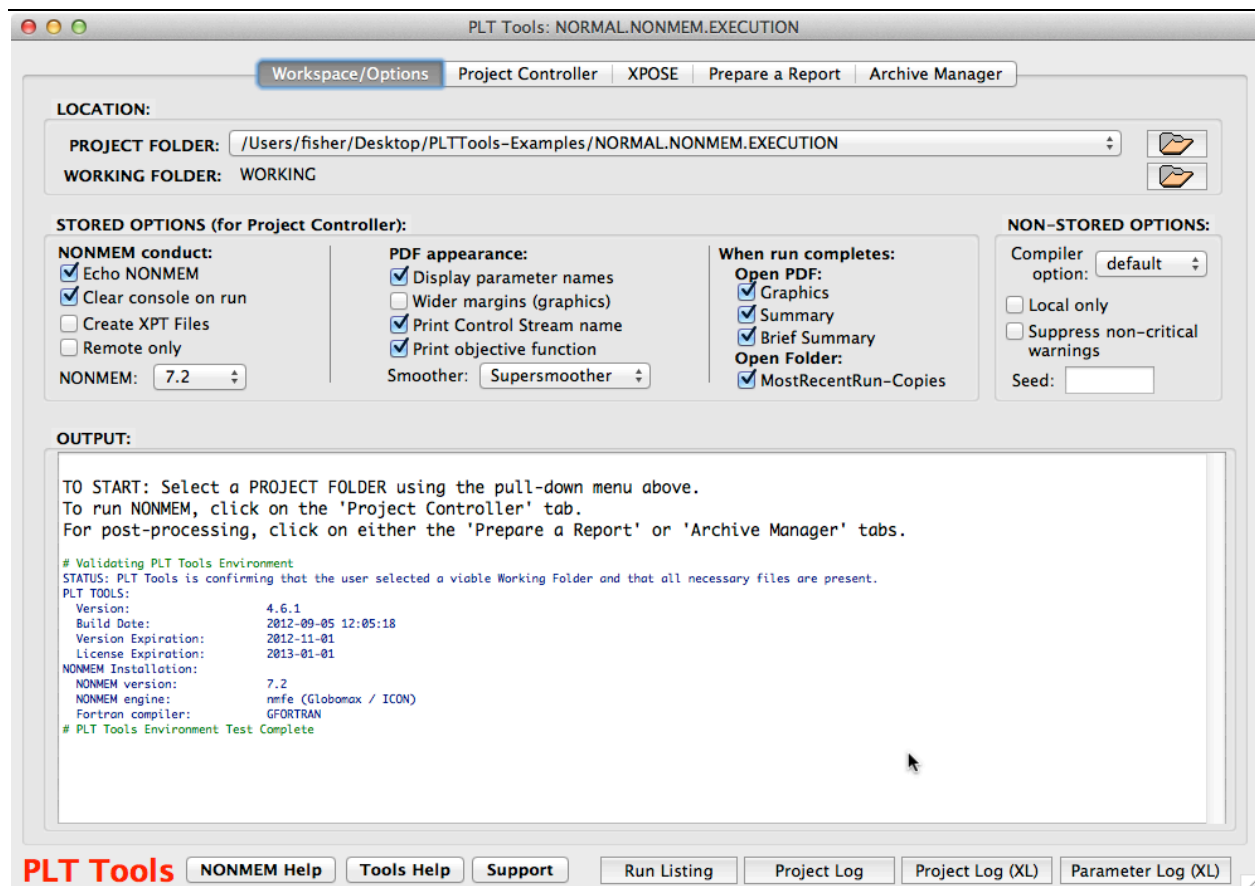


Figure 13. The Project Folder has been selected.

The Working Folder is within that folder. If the path of the The Project Folder is too long to appear in its entirety, holding the mouse over the field (“hover-help”) displays the entire path.

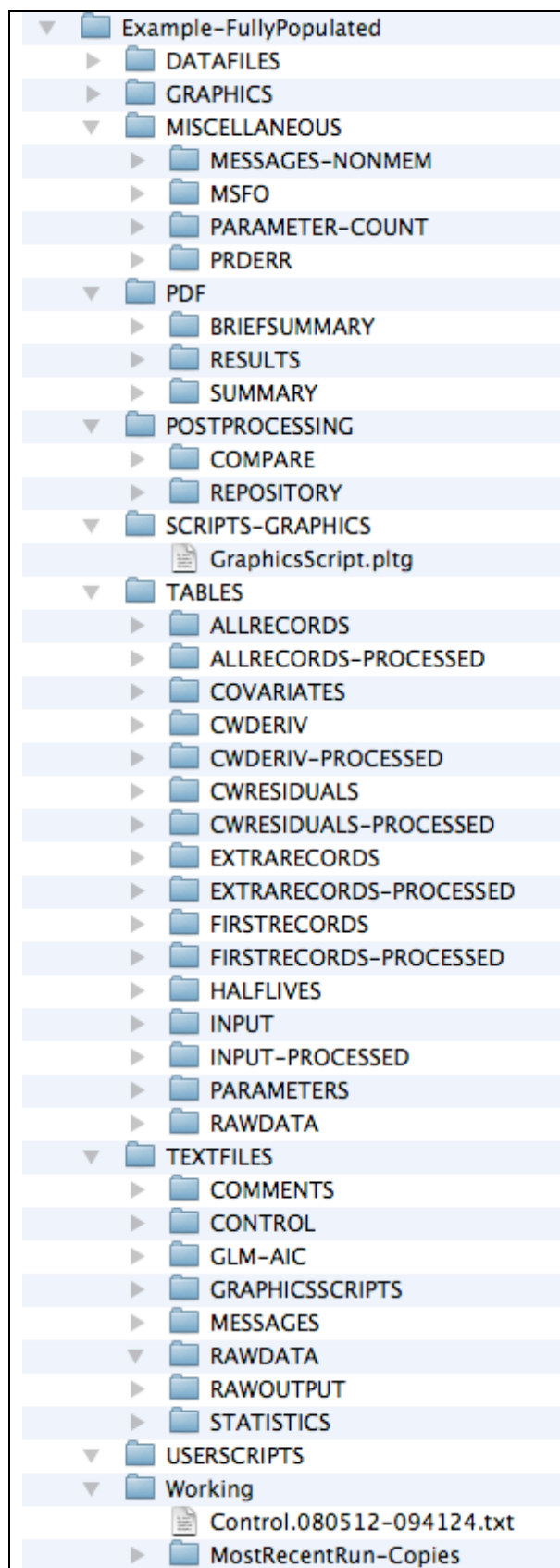


Figure 14. The file structure created by the NONMEM Controller is shown. The display from OS X is displayed.

4. Move to the Project Controller tab (Figure 15).

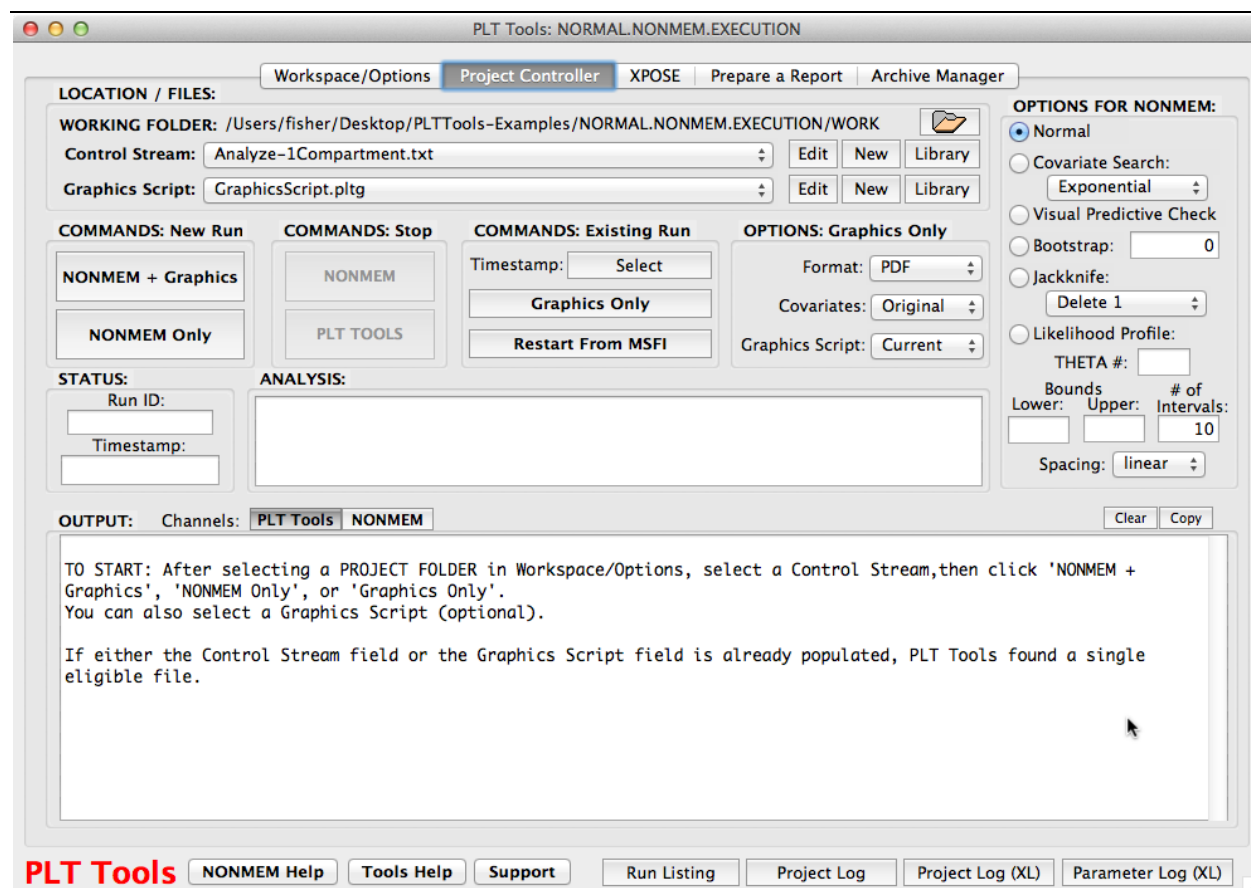


Figure 15. The Projector Controller tab is displayed.

Note that the Working Folder selected in Workspace / Options is displayed.

5. Select a Control Stream. If the user created a Control Stream previously, this Control Stream MUST be in the Working Folder. Use the pull-down menu to select a file. If a Control Stream does not exist, the user can click **New**. A library of Control Streams is provided – the user can select one, then edit the Control Stream for the particular problem. To edit a new or existing Control Stream, click **Edit**.

6. Select a Graphics Script. If a Graphics Script already exists (e.g., one provided with the Examples or one created by the user), use the pull-down menu to select the file (if there is only one file in the Scripts-Graphics folder with the extension .pltg, **PLT Tools** selects that file automatically). If a Graphics Script does not exist, click **New**, then select a name (any name is acceptable — the extension MUST be .pltg). To edit a new or existing Graphics Script, click **Edit**. **This entire step can be deferred. If it is deferred, PLT Tools will create “default graphics”.**

7. Click **NONMEM + Graphics**.

8. If the user has not created the Graphics Script or wishes to modify the graphics after the NONMEM run is complete, graphics can be created again (overwriting the previous version).

First, select the time-stamp for the run for which graphics should be created using the pull-down menu in the “Run #” field (if the user fails to perform this step, **PLT Tools** assumes that the user wishes to create graphics for the most recent run, confirming this with a dialog box). Next, click the **Graphics Only** button; graphics will be created.

If the “Graphics Only” option is selected, all covariates provided at the time of the initial analysis are incorporated into the graphics (the Covariates field reads “Original”). By default, if the user has updated the covariates file in order to incorporate new covariates into graphics, the new covariates file is not read. If the user wishes to use the contents of this revised covariates file, it is necessary to select “Updated” in the Covariates field. If “Updated” has been selected, a message is sent to the Output window confirming that this option has been selected. In addition, a disclaimer is printed on all relevant pages of graphics. This disclaimer indicates that the graphics have been revised but that these covariates may not be incorporated correctly in the NONMEM analysis. **The user can suppress that disclaimer text by creating a file 'OmitCovariatesModifiedText.txt' in the WORKING FOLDER.** The user is responsible for re-running the NONMEM analysis in order to include new covariates.

If the “Graphics Only” option is selected but a Graphics Script has not been selected, **PLT Tools** will indicate that a Graphics Script must be selected before invoking the “Graphics Only” option.

8. If the user wishes to conduct a NONMEM analysis without creating any graphics, click **NONMEM Only** instead.

Details for these steps are provided below.

Text in the Output Window

The Output Window contains two “channels” (superimposed windows); the appropriate window is typically selected by **PLT Tools**. However, the user can select the alternate window using buttons labeled “PLT Tools” and “NONMEM” adjacent to the text “OUTPUT”.

The “PLT Tools” Output window displays text from two sources: **PLT Tools**’ graphical interface, the R code running within **PLT Tools**, and NONMEM. Colors are used to distinguish the source of the text:

PLT Tools (graphical interface): All text is green.

PLT Tools (R code): Normal text is blue. Messages that require attention are displayed in magenta (“INFORMATION”), brown (“WARNING”), or red (“ERROR”); these colors indicate higher priority.

In the “NONMEM” Output window, all text is black.

Text in the Analysis Window

The Analysis window reports the status of a NONMEM analysis. Messages to displayed to indicate when each of the Control Stream and Graphics script have been read. Once these messages have been displayed, these files can be edited as appropriate.

Next, the Analysis window reports the status of NMTRAN, then NONMEM. When NONMEM is complete, the window reports that post-processing (graphics and statistics) is in progress.

During multi-iteration runs, *e.g.*, likelihood profiling, bootstrap, and jackknife, the current iteration is reported in the window. If multiple iterations are being run simultaneously (only available to licensed users who choose the “Parallel Runs” option), the iteration started most recently is reported.

Modifications needed for the NONMEM Control Stream

A few modifications to the Control Stream are needed. These include:

1. In order to standardize file storage structure, **PLT Tools** expects the dataset to be stored in DATAFILES (a folder within PROJECTFOLDER) or a subfolder within DATAFILES. In some instances, this may be impractical, *e.g.*, if the dataset is the output from another run.* **PLT Tools** will send a message

INFORMATION: It appears that the dataset is not located in the DATAFILES folder. PLTsoft recommends that the dataset be located in PROJECTFOLDER/DATAFILES.

To suppress this message, add

`; ALLOW.ALTERNATE.LOCATION`

to the Control Stream.

2. The user must create at least one table in which the FIRSTONLY option is not selected (by not invoking the FIRSTONLY options, the number of **Figure 1** in that table equals the number of records in the input file). This table must be saved as a file and named `AllRecords.txt` or `AllRecords`. The NOPRINT option must be selected. Acceptable syntax for the table record is:

```
$TABLE [LIST OF COLUMNS FOR THE TABLE]
      NOPRINT FILE=AllRecords.txt ; or AllRecords Or AllRecords.lst
```

Recommended columns for this table include ID and TIME; if *post hoc* predictions are obtained, it is mandatory to include IPRED (or the corresponding term). Additional recommended columns are either EVID (preferred) or MDV and AMT (and RATE if relevant). If these are not provided (and if \$PK is used), typically they are accessed from FDATA. However, if \$PRED is used, FDATA may not include critical items. IRES is optional; if not included, it will be calculated during post-processing, assuming that IPRED is provided.

3. The user is encouraged to create a second table in which the FIRSTONLY option is selected. This yields a table with only a single record per subject. This table must be saved as a file and named `FirstRecords.txt` or `FirstRecords`. The NOPRINT option must be selected. Acceptable syntax for the table record is:

* This might occur in the analysis of a metabolite or an effect measure preceded by a pharmacokinetic model. The output from the pharmacokinetic model (possibly an “ExtraRecords” file – see below) might be used as the input for the subsequent analysis.

```
$TABLE [LIST OF COLUMNS FOR THE TABLE]
      FIRSTONLY NOPRINT FILE=FirstRecords.txt ; or FirstRecords
                                           ; Or FirstRecords.lst
```

The only mandatory column for this table is ID. Recommended records are all *post hoc* parameters and all *post hoc etas*.

4. If the user needs to create a third table (e.g., as input to another NONMEM run, naming the file ExtraRecords or ExtraRecords.txt or ExtraRecords.lst will result in the table being renamed and filed using the conventions described above.

5. The user is welcome to insert comments into the Control Stream. Any records preceded by “;C” are interpreted as comments (see COMMENTS section below).

6. If the user creates a “model specification file” (see NONMEM help files), the following text should appear in the \$ESTIMATION record: MSFO = msfo.outputfile. If the user does so, this file will be renamed msfo.TIMESTAMP and will be filed in MISCELLANEOUS/MSFO. If this file has not been requested (and if the "Display Live Parameters" option is checked in Preferences), the required text is added to the Control Stream.

Optional Modification to the NONMEM Control Stream: NMTRAN typically requires that the user enter the names of data items in the \$INPUT record in the Control Stream. In some instances, the user may have identified these names in the first row of the dataset. For example, the first row of the dataset may read:

```
PTID=ID PERIOD TIME CP=DV EVID AMT
```

Under certain circumstances, the user need not complete the \$INPUT record in the usual manner. Instead:

1. The \$INPUT record should read exactly:

```
$INPUT FROMDATASET
```

2. The first line of the dataset should have the exact syntax that is expected in the Control Stream.

a. It should contain the exact names that will be read by NMTRAN / NONMEM. If a version of NONMEM before 7.0 is used, names are limited to 4 characters.

b. Commas and quotes are permitted

c. Aliases are permitted (i.e., PTID=ID); DROP or SKIP are also permitted.

When these conditions are met, **PLT Tools** comments out the original \$INPUT record and inserts a new \$INPUT record consisting of the string \$INPUT followed by the content of the first line of the dataset. Commas are replaced with spaces and double quotes are removed.

When **PLT Tools** inserts the new \$INPUT record, a message is sent to the Output window.

If the user invokes this option, it is important that the IGNORE option in \$DATA in the Control Stream is invoked correctly: do not precede the text in the first line of the dataset with any additional characters (spaces are allowed).

Assume that the Control Stream contains:

```
$DATA ... IGNORE="@"; or IGNORE="P"
```

If the dataset reads:

```
PTID=ID TIME CP ...  
1      0      0
```

the input record in the Control Stream is converted (correctly) to:

```
$INPUT PTID=ID TIME CP ...
```

In contrast, if the dataset reads:

```
#PTID=ID TIME CP ...  
1      0      0
```

PLT Tools does not strip off the "#" character from the first line of the dataset. As a result, the input record in the Control Stream converts to:

```
$INPUT #PTID=ID TIME CP ...
```

and NMTRAN/NONMEM fails.

Modifications needed if \$PRED is used instead of \$PK

When NONMEM is run using \$PRED (instead of \$PK), there are a number of differences in the handling of reserved terms. The most relevant example is that \$PRED does not require that the dataset include a TIME data item. When **PLT Tools** processes the output of a NONMEM run that does not contain time, **PLT Tools** cannot find the expected TIME values for certain graphics (e.g., by-subject graphics and spaghetti plots). However, these graphics are often desired with a different variable (e.g., dose or concentration) on the x-axis. In order to accomplish this, **PLT Tools** must be informed that a different variable is to be used on the x-axis. Either of two approaches can accomplish this:

1. In \$INPUT, use the term TIME instead of the appropriate term (e.g., DOSE). NONMEM does not assign special meaning to TIME in this setting.

2. In the \$PRED block, add a statement:

```
TIME = DOSE; the right-hand side must contain a data item from  
$INPUT
```

With both approaches, add TIME to the \$TABLE statement. When **PLT Tools** processes a table containing time, it will display the appropriate graphics. The default label on the x-axis will be "Time (units)". This can be replaced with appropriate text — see *"Replace 'Time' as the X label"*.

Modifications to the Dataset

PLT Tools does not require any modifications to the dataset. However, one optimal modification (see next section) may improve the quality of graphics.

Updating Initial Estimates Based on the Output from a Previous Run

There are circumstances in which, after completing a NONMEM run, the user revises the initial parameter estimates based on the results of a previous run. For example, an initial run may have been performed based on crude estimates, in which case, applying better starting estimates might improve run-times significantly. In addition, simulation (including conduct of a

visual predictive check) requires imposing initial estimates from a previous run. The user could enter these values manually based on text or PDF outputs. However, to facilitate this step, **PLT Tools** formats parameter estimates (*THETAS*, *OMEGAS*, *SIGMAS*) from each run. These values are stored in a file `FormattedParameters.TIMESTAMP.txt` in `TABLES/FORMATTED.PARAMETERS`. A copy appears in `MostRecentRun-Copies`. The user can edit the Control Stream (accessed most readily from the `EDIT` button in **PLT Tools**, then paste the new text into the Control Stream. The user can choose to incorporate all of the parameter estimates from the previous run or a subset.

If all *OMEGA* and *SIGMA* elements are diagonal (*i.e.*, no covariance is permitted between any of these terms), the file should reflect the parameter estimates from that run exactly. If there are off-diagonal elements for either *OMEGA* or *SIGMA*, **PLT Tools** attempts to format these exactly; however, the user is requested to verify accuracy. All 0 values are followed by the text `FIXED` (because 0 can be an initial estimate only if the value is fixed).

Adding Records to the Dataset to Smooth PRED/IPRED vs. TIME CURVES

There are many circumstances, such as the occurrence of sparse sampling or repeated steady state dosing, that a display of `PRED` or `IPRED` vs. `TIME` does not accurately portray the concentration over time. To accurately portray the time course of concentration requires adding records to the dataset in which either `MDV=1` (no dependent variable) or `EVID=2` ("other type" record), or `AMT=0` (no dose). For the added records `TIME` should be spaced relative to each dose in order to provide enough predictions to show to accurately portray the concentrations after the dose. The user can add these records during creation of the dataset. As an alternative, **PLT Tools** can add these records before running `NMTRAN/NONMEM`. Two approaches are available:

Approach 1:

The Control Stream must contain the following text on a line by itself:

```
; PLTTOOLS:      EXTRADATARECS | NUMBER OF RECORDS | DURATION
```

or

```
; PLTTOOLS:      EXTRADATARECS | NUMBER OF RECORDS | DURATION | LOG
```

where `NUMBER OF RECORDS` is an integer value indicating the number of additional records to be added for each dose, `DURATION` is the number of time units over which these values will be spaced, and `LOG` (which is optional) indicates whether the time intervals should be log-spaced. Note that the entries are separate by vertical bars.

For example:

```
; PLTTOOLS:      EXTRADATARECS | 6 | 30
```

places extra records at 5, 10, 15, 20, 25, and 30 time units after each dose, and

```
; PLTTOOLS:      EXTRADATARECS | 6 | 16 | LOG
```

places extra records at 0.5, 1, 2, 4, 8, and 16 time units after each dose.

Approach 2:

The Control Stream must contain the following text on a line by itself:

```
; PLTTOOLS:      FORCEDATARECS AA|BB|CC
```

where AA, BB, CC, *etc.* are real numbers. If only a single extra record per dose is required, the text would be:

```
; PLTTOOLS:      FORCEDATARECS AA
```

Negative values are allowed. However, if a negative value results in a time entry < 0 , NONMEM will abort (except version 7.4 of NONMEM, which supports time < 0).

With both approaches, to insert the extra records, **PLT Tools** must be able to interpret several columns in the data set. Specifically, the dataset must:

- a. be in CSV format
- b. the first line of the dataset must be the column names
- c. column names must include: ID, TIME, AMT, DV.

Except for ID, aliases are NOT permitted in these entries (aliases can be used in the Control Stream in the normal manner). If the ID column is the *first* column, the header for that column can be replaced with PLTID or PLTID=ID. This is useful if the user attempts to open the CSV file with Excel (Excel balks, and may crash, if cell A1 in a spreadsheet is ID).

- d. column names must contain either (or both) EVID or MDV. This is necessary because the new records are flagged with EVID=2 or MDV=1.
- e. if EVID, MDV, CMT, ADDL, RATE, II data items are present in the dataset, they must not be aliased in the names that appear in the dataset.
- f. under certain circumstances, the user might include multiple EVID columns in a dataset to permit selection of different sets of data (*e.g.*, to include / exclude BQL values). These can be named EVID0, EVID1, and/or EVID2. However, at least one should be named EVID.

When these conditions are met, the following steps occur:

1. **PLT Tools** reads the dataset.
2. A duplicate copy is made of every record with AMT > 0 .
3. Certain data items in these records (if present) are adjusted: EVID=2, MDV=1, RATE=0, ADDL=0, II=0.
4. AMT is set to zero for the duplicated AMT records.
5. if there is a CMT data item in the dataset, CMT values in these records are replaced with CMT values associated with observation records (EVID=0 or MDV=0). If there are observations in more than one CMT, the entire set of records is replicated as needed and the appropriate CMT values used.
6. For each of these records, new records are created in which time is incremented by values determined from the NUMBER OF RECORDS and DURATION. If LOG is not invoked, additional records are spaced by DURATION / NUMBER OF RECORDS. If LOG is invoked, additional records are spaced logarithmically. Note that **PLT Tools** rounds the new TIME values to a reasonable number of digits so that the entries do not violate NONMEM's data input rules.
7. The data are sorted by ID and TIME. The new dataset is used by **PLT Tools**.

8. A copy of the dataset is created in the WorkingFolder with the name:
 DatasetWithAdditionalRecords.csv

The user can then do the following:

- a. Move this dataset to the DATAFILES folder and rename it as desired (change the Control Stream to use the new file).
 - b. Remove the additional text from the Control Stream (as detailed in the first step above).
- Although these steps are optional, executing them may save a few seconds during startup.

WARNING: PLT Tools is unable to add additional records if the dataset contains DATE=DROP (or DAT1=DROP, DAT2=DROP, or DAT3=DROP).

NEW FEATURE: The user should confirm that adding extra records does not have unanticipated effects on the analysis. One situation in which problems might occur is if a parameter changes as a result of a covariate in the data. For example, consider a model in which clearance changes at the time identified by a data item named FLAG. A subset of a dataset before and after modification is shown below; the additional records are italicized. Note that the final two records (at TIME=2, 3) contain FLAG=1. However, in the original dataset, FLAG changed to 2 at TIME=2. In addition to reviewing the dataset after modification (it can be found in PROJECTFOLDER/TABLES/INPUT-PROCESSED), the user should compare runs that differ only by the addition of the extra records: the objective function and parameter estimates should be identical in the two runs (with the possible exception of rounding errors).

Original Dataset:

TIME	EVID	FLAG
0	1	1
2	1	2

With Additional Records:

TIME	EVID	FLAG
0	1	1
<i>1</i>	<i>2</i>	<i>1</i>
2	1	2
<i>2</i>	<i>2</i>	<i>1</i>
<i>3</i>	<i>2</i>	<i>1</i>

To address this problem, do the following:

1. Insert the following text in the Control Stream:
 ; PLTTOOLS: TIME.VARYING.COVIARIATES

This text **must** precede

```

; PLTTOOLS:      EXTRADATARECS
; PLTTOOLS:      FORCEDATARECS

```

2. Create a text file named:

TIME.VARYING.COVIARIATES.txt

in the WORK folder. This file must contain one entry per line. The entries should be the column names (one or more) in the input file that require repair. In the example above, there would be a single entry FLAG. Note that these entries correspond to the column headers in the input file (which might or might not correspond to the names in the \$INPUT statement in the Control stream).

3. **PLT Tools** will identify and repair the issue in the identified above in "With Additional Records".

If the dataset contains an ADDL (and the accompanying II) data item, PLT Tools will attempt to add records relative to every explicit and ADDL dose. If adding records relative to ADDL doses increases the number of records excessively, PLT Tools can be instructed to add records relative to explicit doses only. This is accomplished by adding the following code to the Control Stream:

```
; PLTTOOLS: IGNORE.ADDL
```

This entry must appear BEFORE

```
; PLTTOOLS: EXTRADATARECS
```

Entering LOQ Values in the Dataset

In certain graphics, a dotted line displays the assay limit of quantification (LOQ) value. In addition, in BySubject graphics, BQL values that are used in the analysis are displayed differently from those that are not used in the analysis.

Normally, LOQ is provided to **PLT Tools** in the Graphics Script in the "General Layout / Options" tab.* This is appropriate if a single LOQ value applies to all samples. If the dataset includes two dependent variables (e.g., Cp and an effect or parent and metabolite), different LOQ values for each can be entered into a Graphics Script.

However, there may be more than one LOQ value for a dependent variable, e.g., a result of different assays. To address this issue, **PLT Tools** allows the user to create a data item LOQV. If the dataset contains a column named LOQV and this data item is output to the AllRecords file (alternatively, an LOQV data item could be created in NONMEM code, then output to the AllRecords file), **PLT Tools** applies the LOQ values in this column in graphics. Presence of an LOQV entry in the AllRecords file overrides entries in the Graphics Editor, applying the value obtained from the LOQV data item.

Conditional Weighted Residuals (CWRES)

Prior to version 7 of NONMEM, calculation of conditional weighted residuals required special code in NONMEM. In NONMEM 7, this special code is no longer required. This section refers to user-defined calculation of conditional weighted residuals, as required in NONMEM 5 and NONMEM 6. Text that follows also applies to NONMEM 7 if the user inserts the code shown below.

PLT Tools supports calculation and display of conditional weighted residuals (Hooker *et al.* *Conditional weighted residuals, an improved model diagnostic for the FO/FOCE methods*. PAGE 15 (2006) Abstr 1001 [<http://www.page-meeting.org/?abstract=1001>]) based on code provided by Hooker (xpose.sourceforge.net/generic_chm/compute.cwres.html, with permission from Hooker).

* If the dataset includes two dependent variables (e.g., Cp and an effect or parent and metabolite), different LOQ values for each can be entered into a Graphics Script.

The user is encouraged to review the extensive documentation of CWRES provided at http://xpose.sourceforge.net/generic_chm/compute.cwres.html. In order for calculate CWRES using **PLT Tools**, minor modifications are required to the procedures described by Hooker *et al.*

Recommended by Hooker:

In the \$PRED block:

```
OPEN(50,FILE='cwtab1.est')
```

In the \$TABLE block:

```
$TABLE ID COM(1)=G11 COM(2)=G21 COM(3)=G31 COM(4)=H11 COM(5)=H21  
IPRED MDV NOPRINT ONEHEADER FILE=cwtab1.deriv
```

Necessary for PLT Tools:

In the \$PRED block:

```
OPEN(50,FILE='cwtab.est')
```

In the \$TABLE block:

```
$TABLE ID G11 G21 G31 H11 H21  
IPRED MDV NOPRINT ONEHEADER FILE=cwtab.deriv
```

PLT Tools searches the Control Stream for the presence of `cwtab.est` and `cwtab.deriv`. If these files are created, **PLT Tools** renames them for archival purposes, then attempts to calculate CWRES. Despite the presence of these tables, there are circumstances in which CWRES cannot be calculated (*e.g.*, if the user outputs incorrect data to `cwtab.deriv`). If CWRES cannot be calculated despite the presence of the tables `cwtab.est` and `cwtab.deriv`, a warning message is sent to the Output window. If CWRES are calculated successfully and graphics are created, a set of graphics displays CWRES vs. various other metrics.

New Methods in NONMEM

NONMEM 7 introduced a number of new estimation methods beyond FO and FOCE. A major change is that the user can run sequential methods, *e.g.*, iterative two-stage, followed by SAEM, followed by FOCE; this required some restructuring of outputs within **PLT Tools**. **PLT Tools** accommodates all these methods:

1. **PLT Tools** runs “generic” NONMEM, *i.e.*, NONMEM is executed with a command similar to this:

```
nmfe72 CONTROLSTREAM OUTPUTFILE
```

As a result, any NONMEM control stream that runs outside of **PLT Tools** (*e.g.*, at the command line or in Wings for NONMEM) should run within **PLT Tools** (once **PLT Tools** has been configured initially). If a Control Stream runs outside of **PLT Tools** but not within **PLT Tools**, the most likely explanation is that the user has invoked different versions of NONMEM or different Fortran compilers.

2. When new NONMEM methods are used, NONMEM's output differs. This is most obvious when one uses several sequential methods: each of the methods will yield a set of parameter values and the minimum value of the objective function. The **PLT Tools** output file that is

affected is the Brief Summary. That file has been adapted so that it includes each set of parameter estimates, identified by the method.

3. When more than one method is used, there may be more than one value for the objective function. To prevent confusion between these values, they are no longer displayed in the Graphic file (the option `PDF Appearance: Print objective function` is ignored).
4. Graphics are otherwise unchanged. These graphics are based on the output tables.

Note: An option exists to direct the output from an estimation step to a designated file (`FILE=filename`). If the user selects this option and overrides the default output file (`root.ext`, where `root` is the name of the control file, not including any extension), **PLT Tools** directs a message to the console recommending against that action. **NONMEM** automatically stores these outputs in the file `root.ext`. Directing them to another file deprives **PLT Tools** of access to those data. In addition (depending on the filename selected by the user), **PLT Tools** may not be able to rename / reformat / archive the output file.

Note: Although extensive testing has been performed to assure that **PLT Tools** accommodates the new methods, unidentified issues may arise. If the user encounters errors or identifies ways in which the output could be improved, please contact support@PLTsoft.com

NONMEM's CHECKOUT Option

The `CHECKOUT` option in `$DATA` permits `NMTRAN` to read the user's dataset, then output tables. This option is useful to evaluate whether a dataset is formatted correctly. If this option is invoked, **PLT Tools** will use the tables to create default graphics or, if a Graphics Script has been created, graphics to the user's specification. These graphics can aid in evaluating if the dataset contains errors.

A Control Stream (`PLTTools-DataCheckout.ct1`), specifically tailored to permit use of the `CHECKOUT` option, is provided in the Control Stream Library (which can be accessed from the `File -> Open` menu). The Control Stream is structured to take advantage of **PLT Tools'** `FROMDATASET` option for `$INPUT` (this option replaces the text `FROMDATASET` with column names obtained from the first record of the dataset). If the dataset is not set up with the column names in the first row of the file, the user can enter data items for each of the `$INPUT` and `$TABLE` records in the Control Stream. Data item aliases are permitted, provided that they are syntactically correct. For example, an entry `DV=Cp` is handled correctly.

The user can also write their own Control Stream.

Providing Covariate Data

Covariate data (e.g., demographics, laboratory values) can be provided using either of two approaches:

1. A separate file, delimited either by tabs, commas, or semicolons, can be created by the user. This file should contain a single line per subject and must include column headers. This file can

be created using Excel, then invoking the “Save As” option to select a delimiter. This is the preferred approach.

2. The user can incorporate the covariate data into the NONMEM dataset. If this approach is selected, the `AllRecords` table (see previous section) must include all covariates that the user wishes to examine graphically. Because of limitations in NONMEM on both the number of columns that can be read into NONMEM and the number of columns in a table, this approach is not recommended.

Starting a New NONMEM Run

To run NONMEM, move to the Project Controller tab and select a Control Stream using the **Select** button.

If there are more than 20 Control Streams (files ending in `.pltc`, `.ctl`, `.con`, `.txt`, `.text`) in the `WORKINGFOLDER`, only 20 will be displayed; additional files can be accessed using the “MORE...” option. However, **PLT Tools** discourages having more than 20 Control Streams in the `WORKINGFOLDER` and encourages that excess files be moved to a folder `ADDITIONAL.CONTROL.STREAMS` that it creates in `WORKINGFOLDER`. The limit of 20 files can be revised by creating a textfile:

```
NControlStreams.txt
```

in the folder `PLTTools-Support/CONFIGURATIONFILES`. This file should contain a single entry – an integer indicating the new limit. If the file exists but is not constructed correctly, a value of 1000 will be applied.

New Control Stream

The user can also create a new Control Stream, based on a series of templates. This is accomplished by clicking the **New** button. This opens a dialog box showing templates for a number of common Control Streams (ADVAN 1, 2, 3, 4, 11, 12). Once the user selects a Control Stream, a dialog box asks the user to name the file: the user can accept the default name or assign a new name. The Control Stream is saved automatically in the Working Folder.

Editing the Control Stream

The user can also edit the Control Stream, regardless of whether the file was created by the user or was one of the templates provided with **PLT Tools**. **PLT Tools** contains a simple built-in editor, sufficient for most purposes. However, the user can select an external editor in Preferences. To use the editor, click the **Edit** button; when editing is complete, the file is saved with platform-appropriate line endings.

Graphics Script

Next, select a graphics script (described below) using the **Select** button. If a graphics script is not selected, default graphics will be created at this time; later, the user can create/select a graphics script to obtain customized graphics.

Finally, click the button **NONMEM + Graphics**. After **PLT Tools** confirms that the selected files exist, **PLT Tools** will attempt to run NONMEM. If the Control Stream and datafiles are syntactically correct and the input file exists, NONMEM will start; messages will be sent to the Output window (the large window at the bottom of the Project Controller). If the user wants to perform NONMEM but not to obtain graphics, click the button **NONMEM Only**. NONMEM can terminate under any of three conditions:

1. The run never started, in which case `outfile` is not created. **PLT Tools** sends a message indicating this and takes no further action.
2. The run starts, `outfile` is created but the `AllRecords` table is not created (nor other tables, if requested). **PLT Tools** performs its tasks to create files but cannot make graphics.
3. The run starts and both `outfile` and the `AllRecords` table are created. **PLT Tools** performs its tasks to create files and graphics. Next, if a graphics script has been identified, **PLT Tools** determines if that file exists, then attempts to create graphics. If the graphics script does not exist, an error message is sent to the Output window (in which case, graphics can be created after a graphics script is created). If the graphics script exists, an attempt will be made to create graphics; if this attempt fails, error messages will be sent to the Output window.

Restarting a Terminated NONMEM Run

With NONMEM 7 or later, the user can terminate a NONMEM run in an orderly manner by clicking the "STOP NONMEM" button (see below). If an `msfo` (model specification file) was created during the earlier run, the run can be restarted at its termination point by clicking the **Restart from MSFI** button. If the user has selected a run number using the **Run #** pull-down menu, **PLT Tools** will restart that run (note that only one "restart" is supported). If the user has not selected a run number, the user will be prompted whether **PLT Tools** should use the most recent run.

Restarting a run requires existence of the `msfo` file. Whether or not the user has requested this file, in most cases, it is created (see **Parameter Update**). The only circumstance in which the file is not created routinely is if the user has not requested its creation and the user has selected to not display real-time parameter updates.

This is a new feature of **PLT Tools**. Although it has been tested extensively, there may be circumstances where it may fail without providing an informed explanation. If this happens, please contact support@PLTsoft.com so that we can improve its function. Note that this feature will not work with NONMEM runs performed before **PLT Tools** version 4.0.1.

Re-creating Graphics from a Completed NONMEM Run

If the user has run NONMEM previously and wants to update graphics related to a particular run, click the button **Graphics only**. If the user has selected a run number using the **Run #** pull-down menu, graphics will be created for that Run Number. If the user has not selected a run number, the user will be prompted whether **PLT Tools** should use the most recent run.

Stopping NONMEM / PLT Tools

Conditions may arise in which the user wants to stop a NONMEM run and/or **PLT Tools**. For example, the user may recognize an error in the Control Stream. There are two means to accomplish this, depending on the user's intent:

1. Stop a NONMEM run but allow NONMEM to exit gracefully: Click the **STOP NONMEM** button. NONMEM resets MAXEVALS to a small number and terminates gracefully, including creating output files and tables.

This button is active only with NONMEM 7.

2. Stop a NONMEM run and stop **PLT Tools**: Click the **STOP PLT TOOLS** button. Both NONMEM and **PLT Tools** terminate immediately. **PLT Tools** then attempts to clean-up residual files.

The run can be restarted at its termination point using the **Restart from MSFI** button (described above).

Simultaneous Runs on a Single Machine or Across a Network.

PLT Tools can perform simultaneous runs on a single machine or multiple runs across a network. Either of these requires a license.

Batch (Sequential) Processing: **This is the only mode that is available for non-registered users.**

If the user initiates a NONMEM run while another NONMEM run is in progress, **PLT Tools** delays the start of the second NONMEM run until the first one is complete. This permits the user to stack NONMEM runs, *e.g.*, to permit sequential NONMEM runs overnight. To accomplish this, **PLT Tools** creates a file named "NONMEMisRUNNING.txt" in the PLTTools folder at the start of each run; at completion of the run, the file is deleted. When a NONMEM run is initiated, if a file with that name exists, the run is delayed until that file is deleted. Under certain circumstances, the file NONMEMisRUNNING.txt may not be deleted when a run completes (*e.g.*, if the run is terminated by the "Stop" button in the Controller window). In these instances, the user is informed that NONMEM may, in fact, not be running and guidance is provided to delete the file NONMEMisRUNNING.txt

Simultaneous Processing: The user can configure a computer to permit two or more simultaneous NONMEM runs. This is advantageous if a multiprocessor computer assigns the runs to separate processors, thereby increasing throughput. To prevent NONMEM from overwriting files, analyses are performed in a temporary set of folders (created by **PLT Tools**). This mode of operation requires registration.

Network Processing: If the user has access to a network in which remote machines are configured to run NONMEM (minimal configuration would require NONMEM + a Fortran compiler + a PLTTools folder), **PLT Tools** can query the network to determine which machines are presently not running NONMEM, then copy appropriate files to the remote machine, run NONMEM, then complete post-processing on the local machine. To prevent NONMEM from overwriting files, analyses are performed in a temporary set of folders. This mode of operation

requires registration plus a supplemental fee based on the number of networked computers. In addition, certain configuration is required: ssh (secure shells, an open-source provide to permit secure communication between computers) must be installed and a user account must exist on each computer to be accessed by the user. The desktop machine can operate either Windows or OS X and the remote machines can be any combination of Windows, Linux, or OS X. It is likely that remote machines running other variants of Unix (*e.g.*, Solaris or BSD) are compatible; however, this has not been tested.

Modes of Operation

Normal

In the normal mode, **PLT Tools** functions identically to running NONMEM at a Command Prompt (Windows) or in a Terminal window (OS X).

Covariate Search

PLT Tools can perform automated univariate covariate searches (*i.e.*, testing one covariate at a time) on any number of covariates and any number of THETAs. Before selecting Covariate Search, the user must populate a Graphics Script, indicating which covariates and THETAs to search. In addition, the user must select whether the new THETA selected for the Covariate Search is applied as Exponential, Multiplicative, or Additive (explained below); the default is multiplicative.

Covariates are referred to hereafter as COVAR1, COVAR2, *etc.*

The Covariate Search procedure is conducted as follows:

1. The first run does not involve any additional covariates. This is the “reference” run to which comparisons are made.
2. After the first run is complete, **PLT Tools** uses the COVARIATE and THETA fields in the Graphics Script to determine the covariates and THETAs to search. Then, **PLT Tools** evaluates which of these selected covariates and THETAs are available for analysis. Covariates in the dataset that have been dropped (omitted) using the DROP option are reintroduced into the dataset (“undropped”) at the appropriate run (see Step 3 below).*
3. If the covariate to be evaluate has been dropped, the Control Stream is edited to delete one instance of “=DROP” code; NONMEM is then re-run. This run is otherwise identical to the reference run except that the FDATA file created by NMTRAN now contains the “un-dropped” data column.

* One exception applies: NONMEM 5 and 6 restrict the number of input columns to 20 (19 if MDV is not one of these columns. If the dataset already includes the maximum number of allowable columns, “undropping” a column will cause the number of input columns to exceed the limit, creating an error condition in NONMEM. If this happens, the user must drop another column to allow a Covariate Search to be performed.

4. The FDATA file created in step 1 (or step 3) is examined to determine the first value of COVAR1 for each subject. The median value of this covariate is determined (so that the analysis can be centered).

5. The Control Stream is modified to apply an effect of the covariate to a single THETA. For example, consider a Control Stream that includes the text:

```
CL = THETA(1) * EXP(ETA(1))
```

The Control Stream is modified differently depending on the search mode selected by the user (added text is underlined):

Multiplicative:

```
CL = (THETA(1) * (1 + (THETA(X) - 1) * (COVAR1 - MEDIANCOVAR)) * EXP(ETA(1)))
```

Exponential:

```
CL = (THETA(1) * (COVAR1 / MEDIANCOVAR) ** (THETA(X) - 1)) * ETA(EXA(1))
```

Additive:

```
CL = (THETA(1) + (THETA(X) - 1) * (COVAR1 - MEDIANCOVAR)) * ETA(EXA(1))
```

where MEDIANCOVAR is the median value of that covariate and THETA(X) is a new covariate (X is replaced by the 1 + the number of THETAs appearing in the \$THETA block in the Control Stream [e.g., if the Control Stream contained 5 THETAs, X equals 6]). In addition, an initial estimate for the new THETA is added to end of the \$THETA block of the Control Stream:

```
$THETA (1); Additional THETA for the covariate
```

Several aspects of the code are worth noting:

- a. THETA(X) is always offset by 1. This permits the initial estimate for this THETA to be 1; in turn, when the covariate is applied in NONMEM's first iteration, it has no effect (1 - 1 = 0). This approach was taken because NONMEM does not normally allow an initial estimate equal to 0.
- b. The covariate effect is centered either by division (COVAR1 / MEDIANCOVAR) or subtraction (COVAR1 - MEDIANCOVAR). As a result, the parameter estimates apply to the "typical" subject.

6. NONMEM is run using the modified Control Stream. Results are appended into a table (CombinedParameters.TIMESTAMP.csv) and summarized in a graphic (CovariateSearch.TIMESTAMP.pdf).

7. The covariate is applied to each additional THETA; steps 5 and 6 are repeated.

8. The next covariate is tested. If the covariate was "dropped" previously, steps 3 and 4 are repeated to obtain the median value for that covariate. Next, steps 5, 6, and 7 are repeated.

9. After all runs are complete, the summary graphic (CovariateSearch.TIMESTAMP.pdf) is opened.

Issue related to setting up the dataset:

1. If a covariate contains both non-positive values and positive, the exponential approach fails (raising a non-positive or infinite number to a power will generate an error condition in NONMEM).
2. In order to center covariates correctly, a value for each covariate for each subject is obtained from the first record in the dataset. If that value is omitted (*e.g.*, if a NULL ["."] is entered), **PLT Tools** will be unable to perform its task.

Issue related to setting up the Control Stream: If a covariate is “dropped” from the dataset, “undropping” that covariate requires an additional NONMEM run (step 3). If “undropping” covariates does not cause the length of \$INPUT to exceed the number of allowable records (NONMEM 5 and 6: 19 + MDV), “undropping” speeds the procedure.

Visual Predictive Check

PLT Tools provides complete support for performing a Visual Predictive Check. Click on the button to select this option. The Control Stream must contain instructions for simulation (\$SIM) rather than estimation (\$EST). Typically, the SUBPROBLEMS option is set to > 1.

PLT Tools creates graphics tailored for a Visual Predictive Check. These graphics can be tailored for specific features of the dataset, *e.g.*, separate graphics for each of multiple dependent variables, for each of different “periods”, or for each of several groups (such as dose groups). See “Additional Graphics” for details.

Likelihood Profile

PLT Tools provides complete support for performing a likelihood profile for individual parameters. The user must do the following in the Project Controller (**Figure 16**):

1. Select the option **Likelihood Profile**.
2. Enter the THETA to which the Likelihood Profile is applied.
3. Enter the lower and upper bounds over which the Likelihood Profile will be performed.
4. Enter the # of intervals for the Likelihood Profile.
5. Select which the intervals of the Likelihood Profile are spaced in a linear or log manner.

Changes to the Control Stream and Dataset:

One change is needed in the Control Stream: the text

PLTTOOLS LIKELIHOODPROFILE

must appear as a comment (*i.e.*, preceded by a semicolon) on the line in which the initial estimate for the relevant THETA is displayed. That THETA must appear on a line by itself. For example, any of the following would be acceptable code:

```
0.001      ; PLTTOOLS LIKELIHOODPROFILE
(0 0.001) ; PLTTOOLS LIKELIHOODPROFILE
(0 0.001) ; Effect of CrCl on CL ; PLTTOOLS LIKELIHOODPROFILE
```

The following code is NOT acceptable (because two THETAs are displayed on a single line):
 2.0 0.001 ; PLTTOOLS LIKELIHOODPROFILE

No changes are needed for the dataset.

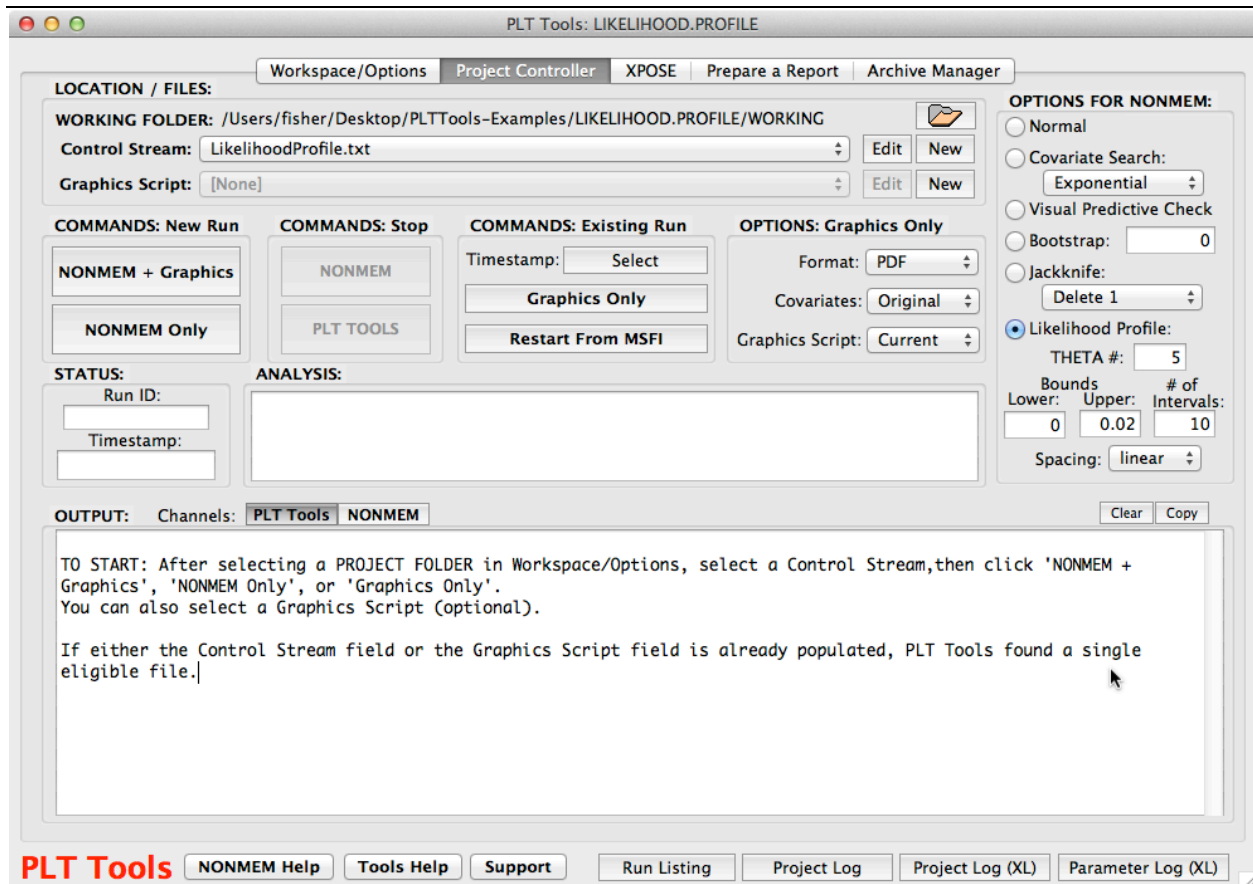


Figure 16. The options fields are configured for a Likelihood Profile.

When the user clicks **NONMEM + Graphics**, the following steps occur:

1. The estimation step is run in the normal manner without fixing the parameter.
2. Based on the information provided by the user, **PLT Tools** creates an array of values to which the parameter is fixed. For example, if the user selects a linear distribution, 10 intervals, and lower and upper bounds of 0.0 and 0.02 for the parameter, the array consists of the values: 0, 0.002, 0.004, 0.006, 0.008, 0.010, 0.012, 0.014, 0.016, 0.018, and 0.020. The estimation step is run 11 times, fixing the value of that THETA to each of these eleven values. This is accomplished with minor modifications to the Control Stream (which are annotated).
3. NONMEM is run using the revised Control Stream. The 13-character ID (see next section) is modified to include “-LKnnnn” where nnnn is the sequential number of that analysis.
4. Parameters from each run are added in sequence to a file `CombinedParameters.TIMESTAMP.csv` in `LIKELIHOODPROFILE/TABLES/PARAMETERS`. The user can access this file to create summary statistics for the bootstrap process.

5. After the final estimation step is complete, a graphic is prepared summarizing the likelihood profile. A copy of the graphic appears in `MostRecentRun-Copies`.

Selecting the lower and lower bounds for a likelihood profile may be a challenge. If the bounds are too narrow or too wide, it may be necessary to repeat the process. To facilitate this, the array of fixed parameter values (0, 0.002, 0.004, 0.006, 0.008, 0.010, 0.012, 0.014, 0.016, 0.018, and 0.020 in the example above) is re-ordered so that the more extreme values are evaluated first. The array is therefore re-ordered to:

0.000. 0.020. 0.002. 0.018. 0.004. 0.016. 0.006. 0.014. 0.008. 0.012. 0.010

After the first several runs have occurred, the user can review the results (the file `Graphics-LikelihoodProfile.TIMESTAMP.pdf`) to evaluate if the lower and upper bounds are reasonable. If not, the run can be terminated, new bound applied, and the process restarted.

If the base run is identical to a previous base run (*i.e.*, this exact model has been run before), the base run can be skipped with the `SKIPBASERUN` option (see `SKIP BASE RUN` section below).

Bootstrap

PLT Tools provides complete support for bootstrap, including setting a "seed" (in order to reproduce the analysis; see below). To perform a bootstrap, the user must check **Bootstrap** and enter the number of bootstraps in **Number** field the Project Controller window. In addition, the Control Stream must contain a `$EST` record and not a `$SIM` record. Typically, the Control Stream will be identical to one used previously for estimation. No changes are needed for the dataset.

When the user clicks **NONMEM + Graphics**, the following steps occur:

1. The estimation step is first run on the entire NONMEM dataset (not bootstrap).
2. The `FDATA` file created during that procedure is saved by **PLT Tools** and used to create datasets for the bootstrap procedure.
3. The `$DATA` record in the Control Stream is altered to indicate the name of the revised dataset.
4. For each bootstrap, a new dataset is created in the following manner:
 - a. Unique IDs in the original dataset are identified.
 - b. These IDs are sampled with replacement until a new set of IDs is obtained. This new set is identical in size to the original set of IDs.
 - c. A new dataset is constructed based on these new IDs. The original IDs are replaced with a new set of sequential IDs ranging from 1 to the number of IDs (*i.e.*, if there are 20 subjects in the original dataset, a new dataset has 20 subjects, numbered 1:20).^{*} A table showing the

^{*} If the Control Stream includes any statements specific to a subject, *e.g.*,

`IF (ID.EQ.2001) ...`

correspondence between the new and original IDs is stored in a file `BootstrapIDMapping.TIMESTAMP-BSnnnn.csv` (where `nnnn` is the sequential number of that bootstrap (e.g., `-BS0001` for the first bootstrap) in the folder `BOOTSTRAP/TABLES/ID-Mapping`.

d. `NONMEM` is run using the revised Control Stream and dataset. The 13-character ID (see next section) is modified to include “`-BSnnnn`” where `nnnn` is the sequential number of that bootstrap (e.g., `-BS0001` for the first bootstrap).

e. Parameters from each bootstrap are added in sequence to a file `CombinedParameters.TIMESTAMP.csv` in `BOOTSTRAP/TABLES/PARAMETERS`. The user can access this file to create summary statistics for the bootstrap process.

f. During post-processing, the original ID is added back to the processed `AllRecords` file (i.e., the delimited version – see next section).

Stratified Bootstrap

PLT Tools can stratify the bootstrap analysis by a categorical covariate. For example, if a dataset included subjects at two different dose levels, each bootstrap dataset could contain the appropriate number of subjects from each dose level. This can be accomplished in **PLT Tools** by adding the following text to the Control Stream:

```
; PLTTOOLS:      STRATIFY.BOOTSTRAP      STRATIFYCOL
```

The first portion of text (`; PLTTOOLS`) indicates to **PLT Tools** that the remainder of the line is a special instruction. The second portion of the text (`STRATIFY.BOOTSTRAP`) indicates that stratification should be applied. The final portion of the text names the data item over which stratification occurs, e.g., `AGEGROUP`. The three portions of text can be separated by spaces or tabs.

The procedure is implemented in **PLT Tools** in the following manner:

1. **PLT Tools** determines if a covariate file is present. If so, it searches the column headers in that file for `STRATIFYCOL`. Counts for each unique entry are determined.
2. If a covariate file is not present, it searches the `FDATA` file for a data item named `STRATIFYCOL`. If that column is identified, the first record for each subject is extracted.
3. If `STRATIFYCOL` is not located, **PLT Tools** quits and informs the user that the stratifying column could not be identified.

these statements will **not** be executed correctly during a bootstrap. This can be avoided by adding a new column to the dataset that replicates the `ID` column (e.g., `NEWID`) and changing the conditional statement to:

```
IF (NEWID.EQ.2001) ...
```

Although `ID` will be replaced, `NEWID` will maintain its original value. **PLT Tools** attempts to identify situations in which a problem could occur (i.e., text in the Control Stream that appears to be conditional on `ID` or an alias) and direct the user to this solution. However, `PLTsoft` is not responsible for errors that could occur in these circumstances.

4. **PLT Tools** assembled a dataframe containing ID and STRATIFYCOL. Counts for each unique entry in STRATIFYCOL are determined. The new population of IDs for the bootstrap is assembled to match the counts in STRATIFYCOL. For example, if there are 60 subjects in age group 1 and 40 subjects in age group 2, the 60 subjects in age group 1 are sampled (with replacement) to generate a new population of 60 subjects in age group 1. The same process is applied to age group 2, then the two subpopulations are combined. In order to avoid sequence effects, the order of the 100 subjects is randomized.

Alternate Bootstrap Approach

In certain instances, the user may wish to sample a smaller subset of subjects. For example, if there are 36 subjects in a dataset, the user may wish to conduct analyses with random subsets of 20 subjects. This can be accomplished in **PLT Tools** by adding the following text to the Control Stream:

```
; PLTTOOLS:      SUBSAMPLE  nn
```

The first portion of text (; PLTTOOLS) indicates to **PLT Tools** that the remainder of the line is a special instruction. The second portion of the text (SUBSAMPLE) indicates that the alternate bootstrap approach should be applied. The final portion of the text (nn) must be an integer value. The value must exceed zero and must be smaller than the number of subjects in the dataset (if it equals or exceeds the number of subjects in the dataset, an error message will appear). The three portions of text can be separated by spaces or tabs.

If the user selects the Bootstrap option for NONMEM and if the text above is parsed correctly, **PLT Tools** will follow the same procedure as described earlier for the bootstrap analysis, except that the sample size will be the nn value identified above. When the analyses are complete, a summary file (SubsetStatistics.TIMESTAMP.csv) will be created; the original of the file is located in PROJECTFOLDER/BOOTSTRAP/TABLES/SUBSET.STATISTICS (a copy exists in MostRecentRun-Copies). This summary file will calculate statistics for each column (except for columns named ETAn and ETnn) in the FirstRecords file. These statistics include the mean, standard deviation, and standard error, the geometric mean, the geometric standard error, and the geometric coefficient of variation, calculated as:

$$\sqrt{\exp(\text{variance}(\log\text{-transformed values}))-1}$$

If the base run is identical to a previous base run (*i.e.*, this exact model has been run before), the base run can be skipped with the SKIPBASERUN option (see SKIP BASE RUN section below).

Jackknife Analysis

PLT Tools provides complete support for performing a jackknife analysis, including setting a "seed" (in order to reproduce the analysis; see below). To perform a jackknife analysis, the user must check **Jackknife** and select the approach being used using the pull-down menu. Options are:

- Delete 1
- 1/2
- 1/2 - random
- 1/5

- 1/5 – random
- 1/10
- 1/10 – random.

These options determine the fraction of subjects removed during each jackknife run. “Delete 1” removes one subject from each run; the number of runs equals the number of subjects. “1/2” removes the first half, then the last half from each of two runs. “1/5” removes sequential fifths of the subjects from each of five runs. “1/10” removes sequential tenths of the subjects from each of ten runs. “1/2 – random”, “1/5 – random”, and “1/10 – random” are similar to the previous approaches except that the subjects deleted from each run are selected randomly rather than sequentially. With all approaches, each subject is deleted from a single run.

Typically, the Control Stream will be identical to one used previously for estimation. No changes are needed for the dataset.

When the user clicks **NONMEM + Graphics**, the following steps occur:

1. The estimation step is run on the entire NONMEM dataset.
2. The FDATA file created during that procedure is saved by **PLT Tools** and used to create datasets for the jackknife procedure. A complete set of IDs is identified; then, subsets to be used in each jackknife analysis are created using the approach selected above.
3. The \$DATA record in the Control Stream is altered to indicate the name of the revised dataset.
4. For each jackknife run, a new dataset is created in the following manner:
 - a. Data for subjects to be included in the run are not changed.
 - b. For subjects whose data is not to be used in the analysis, observations with MDV = 0 (*i.e.*, those samples to be used in the analysis), MDV is changed to 1 and EVID, if present, is changed to 2. As a result, these subjects do not influence the analysis; however, population estimates are obtained for these individuals based on the typical values and their covariates.
 - c. NONMEM is run using the revised Control Stream and dataset. The 13-character ID (see next section) is modified to include “-LKnnnn” where nnnn is the sequential number of that jackknife (*e.g.*, -LK0001 for the first jackknife run).
 - d. An additional graphic is created showing the fit (observed vs. population predicted). One panel shows data for subjects included in the analysis. A second panel shows data for subjects not included in the analysis. A smoother appears in each panel. A copy of the graphic appears in MostRecentRun-Copies.

Setting a Seed for Bootstrap and Jackknife Analyses

Bootstrap analysis and jackknife analysis with the “random” option both require **PLT Tools** to select random subsets of data. In order to ensure traceability and reproducibility of all analyses, **PLT Tools** provides a means to control this “random” process (**Figure 17**). Two situations exist:

1. The user does not select a seed and there are no previous: The first time this occurs, **PLT Tools** selects the value 1 (and this value is stored). Subsequently, the stored value is read and incremented by 1. For example, the second time this occurs, the value 2 would be applied, then 3, etc. The value is reported in the Output window, in a popup window (at completion of the analysis), and in the output files.
2. The user enters a non-negative integer (small integers are acceptable). **PLT Tools** uses this value as the seed for the random-number generator that is used to select subjects for each bootstrap or jackknife analysis and the value is reported in the output files. As a result, if an analysis repeated with the same seed, the datasets will be identical. This applies even if the analysis is performed on a different platform.

To reproduce an analysis, the user must enter a non-negative integer as shown below.

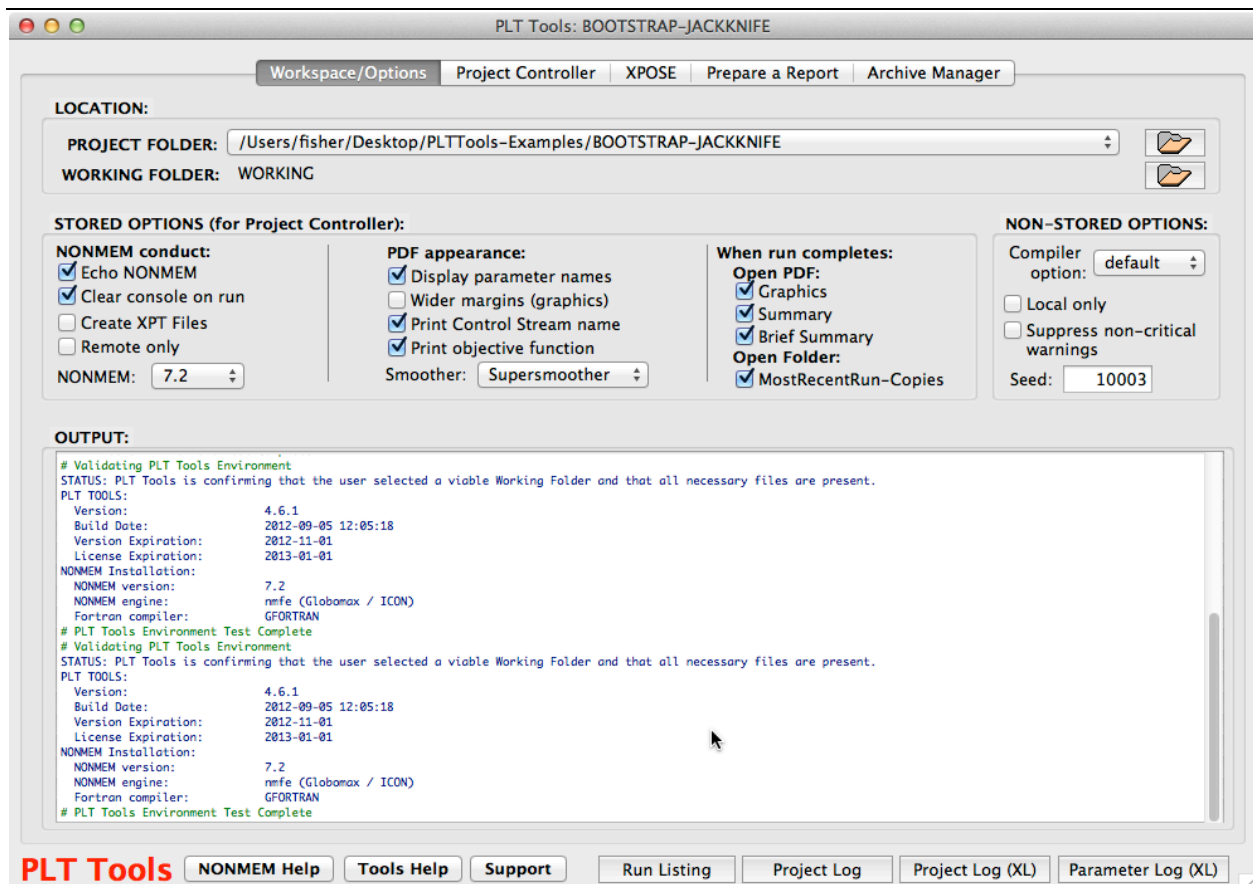


Figure 17. The user can set a seed for bootstrap analyses and certain jackknife analyses by entering a value in the "Seed" field. Only non-negative integers can be entered.

Skip Base Run (Bootstrap Analysis, Likelihood Profile)

When a bootstrap analysis or likelihood profile are conducted, **PLT Tools** first conducts an analysis on the base model. In most instances, this base model has been run before so this step is not needed. The step can be skipped by entering the following text in the Control Stream:

```
; PLTTOOLS: SKIPBASERUN XXXXXX-XXXXXX
```

where xxxxxx-xxxxxx is the 13-character (12-digit timestamp from a prior run of that exact model).

Files Created by *PLT Tools*

Each successful NONMEM run results in the creation of a number of files, each of which is named with a unique 13-character (12 digit) time-stamp (2-digit year, 2-digit month, 2-digit day, dash ("-"), 2-digit hours, 2-digit minutes, 2-digit seconds) indicating the time that the run started. Table 1 documents the location, naming convention, and content of each file. Most text files are preceded by a file header consisting of the time-stamp, a text version of the start and end times of the run, and the names of the computer (the "hostname") and user ("username"). An example of the file header is:

```
TIMESTAMP: 071108-145352
Analysis performed on plessthan2 by fisherdm
Run started at 2007-11-08 14:53:52
Run completed at 2007-11-08 15:16:51
```

Brief Summary

NONMEM output files are often quite lengthy and it is difficult to extract the critical information. This issue has been addressed by the creation of a "brief summary", typically a 1-2 page document that extracts this critical information.

Real-Time Parameter Display

During each NONMEM run, successive iterations are appended to the file INTER. The frequency with which this occurs is controlled by the PRINT option in \$ESTIMATION. The default value is to print the first and last iterations: this approach does not permit display of "real-time" parameter values.

If the PRINT option is assigned an appropriate value (*e.g.*, 1 or 5), the INTER is updated after this number of iterations. **PLT Tools** examines the INTER file to determine when two sets of values (the initial values and one additional set) are available, then a window opens (**Figure 18**) displaying the objective function and each THETA vs. iteration # (only THETAs that are changing are displayed). Each panel also displays the most recent value as text; a time stamp indicates when the display was updated.

If a Graphics Script has been selected for this analysis and if the THETA # column in the Parameters field (tab: Parameters, Etas, Covariates) has been populated, the corresponding labels are displayed in the graphic.

The display can be suppressed or its size changed in Preferences.

At completion of the analysis, a variant of this graphic (formatting more appropriate for a printed document) is saved in PDF format (in the folder PROJECTFOLDER/PDF/TRACKINGGRAPHICS).

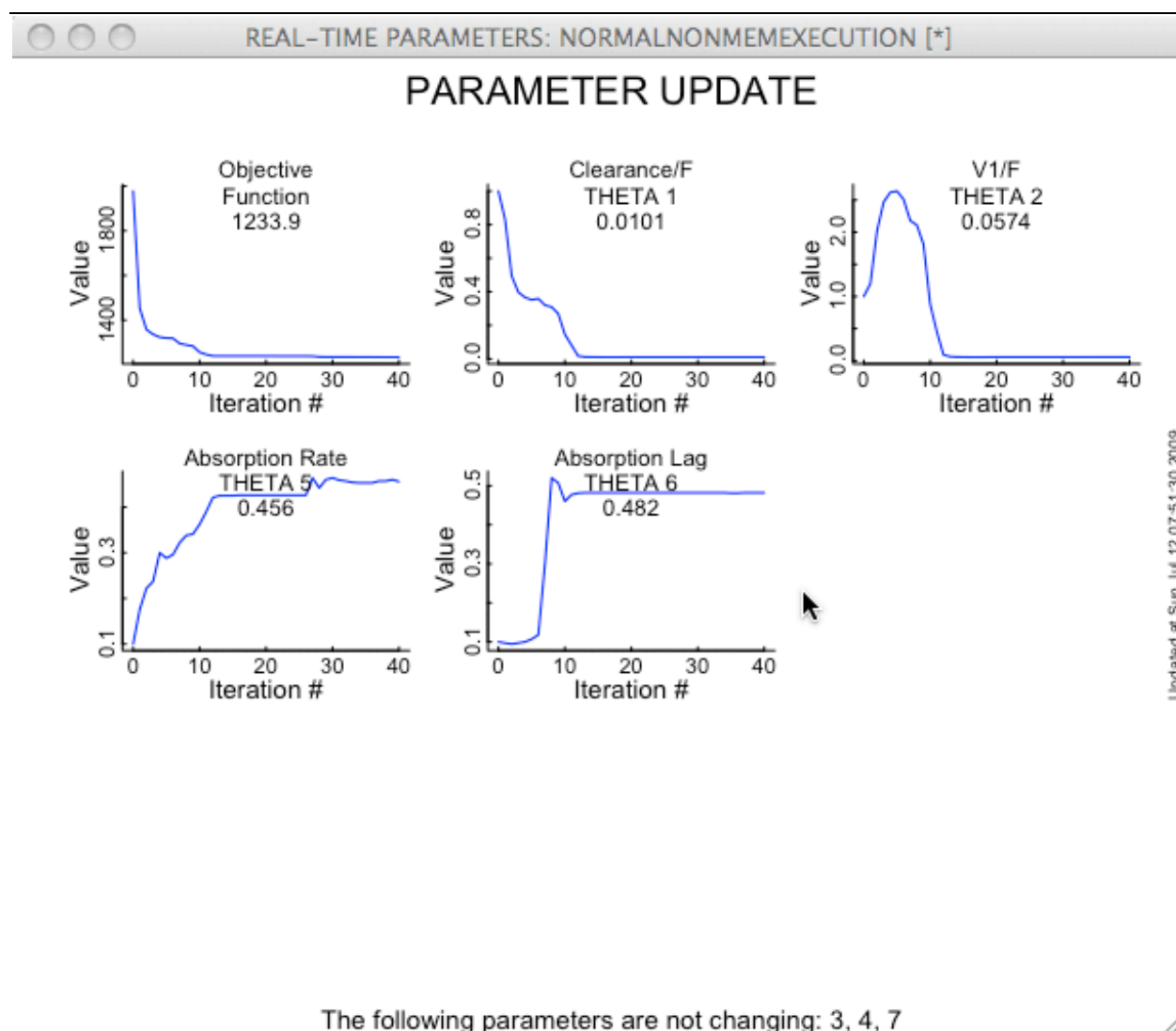


Figure 18. An example of a real-time parameter display is shown.

Embedded Graphics

After a NONMEM run completes, a set of single-panel graphics are prepared in JPEG format. These graphics display observed vs. predicted (population or *post hoc*) values on log scales, observed / predicted (population or *post hoc*) vs. time, *post hoc* η values vs. covariates, and *post hoc* parameter values vs. covariates.

Preservation of Files Created With Each Run

All files created during each run are stored in the appropriate folders within the Project Folder. The user is encouraged to explore these folders to learn of the various files. In addition to these files, up to twenty critical files are copied to a folder, MostRecentRun-Copies, located within the Working Folder. At the completion of the run, this folder opens so that the user can access these files. When the next run is completed,

the contents from the previous run are deleted; as a result, this folder contains only files relevant to the most recent run. In that these files are copies, the user can (but need not) delete these files.

Table 1. Files created by **PLT Tools**. Files that are highlighted are copied to the folder WORKING/MostRecentRun-Copies

Location	
Filename	Content
GRAPHICS Graphics.TIMESTAMP.pdf	Graphics file created from NONMEM tables
MISCELLANEOUS/MSFO Msfo.TIMESTAMP	If an MSFO file named msfo.outputfile is requested in the control file, it appears here.
MISCELLANEOUS/PARAMETERS Parameters.TIMESTAMP.txt	The number of parameters (fixed, non-fixed, total) in the model (<i>thetas, omegas, sigmas</i>) is tabulated.
PDF/BRIEFSUMMARY BriefSummary.TIMESTAMP.pdf	A brief summary of the run appears here.
PDF/RESULTS Result.TIMESTAMP.pdf	The file header precedes the raw output
PDF/SUMMARY Summary.TIMESTAMP.pdf	The output of NONMEM, processed via nmsee.exe is saved in a pdf format.
TABLES/ALLRECORDS AllRecords.TIMESTAMP.txt	The AllRecords table exactly as output by NONMEM
TABLES/ALLRECORDS-PROCESSED AllRecords.TIMESTAMP.csv*	The AllRecords file output by NONMEM with spaces replaced by a delimiter; the first line of the file ("Table # ...") is deleted.
TABLES/FIRSTRECORDS FirstRecords.TIMESTAMP.txt	The FirstRecords table exactly as output by NONMEM
TABLES/FIRSTRECORDS-PROCESSED FirstRecords.TIMESTAMP.csv*	The FirstRecords file output by NONMEM with spaces replaced by a delimiter; the first line of the file ("Table # ...") is deleted.
TABLES/HALFLIVES Halflife.TIMESTAMP.csv*	If "Calculate halflives" is selected in the Graphics Editor, parameters derived for a compartmental model are reported.
TABLES/INPUT Data.TIMESTAMP.txt	The FDATA table exactly as output by NONMEM
TABLES/INPUT-PROCESSED Data.TIMESTAMP.csv*	The FDATA file output by NONMEM with spaces replaced by a delimiter; the first line of the file ("Table # ...") is deleted.
TABLES/PARAMETERS Parameters.TIMESTAMP.csv*	Parameters from a NONMEM run (or multiple sets of parameters from simulation / estimation subproblems) are displayed with column headers
TEXTFILES/COMMENTS Comments.TIMESTAMP.txt	Lines from the Control Stream that are preceded by ";C" appear here
TEXTFILES/CONTROL Control.TIMESTAMP.txt†	The file header precedes the Control Stream
TEXTFILES/GLM-AIC AIC.TIMESTAMP.txt	A detailed summary of the GLM procedure for covariates appears here.
TEXTFILES/GRAPHICSCRIPTS Graphics.TIMESTAMP.txt	The script used to create graphics.
TEXTFILES/MESSAGES Messages.TIMESTAMP.txt	Warnings, errors, and messages collected during creation of the graphics appear here.
TEXTFILES/RAWOUTPUT Output.TIMESTAMP.txt	Raw output from NONMEM

TEXTFILES/STATISTICS
Statistics.TIMESTAMP.txt

This file contains various statistical outputs including medians values of *post hoc* parameter values, *t* tests comparing individual groups, and final results of the GLM procedure.

* The user-selected delimiter (and extension) for these files can be a comma (*.csv*), a tab (*.tab*), or a semicolon (*.csv*).

Define.TIMESTAMP.csv

An additional file named *Define.TIMESTAMP.csv* was added in Version 5.5.2. This file is a template for a *Define.pdf* file that is requested by FDA as part of a regulatory filing. The file contains column names from the input dataset (obtained from the *\$INPUT* record in the control stream). Users can add descriptions to the file, then create the PDF version of the document.

Rescuing Aborted Bootstraps

On occasion, a bootstrap analysis aborts before the requested number of runs have completed. This could result from errors in NONMEM, from problems in the operating system, or from **PLT Tools** encountering unexpected situations. **PLT Tools** allows the user to piece together portions of different bootstraps, thereby recovering all work efforts. This utility will be explained by means of an example:

1. A user wants 500 bootstrap runs. **PLT Tools** aborts after 486 are completed.
2. Select a new seed (in Workspace / Options). When **PLT Tools** performs a new set of bootstraps, the new seed ensures that different subsets of the data are selected.
3. In Project Controller, select the appropriate number of Bootstrap runs. In this case, select 14 (or a larger value). Click **NONMEM** or **NONMEM + Graphics**.
4. When this new run completes, **PLT Tools** will have access to two sets of bootstrap analyses, one containing 486 runs, the other containing 14.
5. In the *WorkingFolder*, create a new text file (extension must be either *.txt*, *.ctl*, or *.con*) with this exact syntax:
 - a. The first line must be:
PLTTOOLS: BOOTSTRAP GRAPHICS
 - b. On subsequent lines, enter the 13-character timestamps for each of the bootstrap runs. A minimum of one entry is required.
 - c. The user can select (sequential) subsets from each of the bootstrap analyses. For example, if the second analysis described above contained 15 runs (for a total of 501), the user could omit one run with the following code:
110706-161739 1 14
where the first part of the text is the timestamp, the second part is the starting position, and the third part is the ending position. Portions of text are separated by a single space.

6. Select this text file as the Control Stream. Select the Graphic Scripts that was used for the bootstrap analyses.
7. Click **NONMEM** or **NONMEM + Graphics**.. **PLT Tools** will bypass most of its usual procedures. Instead, it attempts to do the following:
 - a. read the text file containing the timestamps
 - b. find the bootstrap parameter outputs associated with those timestamps
 - c. merge these files. In particular, **PLT Tools** will search for duplicate sets of parameter estimates (which would occur in the seed was the same for two or more analyses; in theory, it could also happen by change). If this happens, **PLT Tools** will use only unique runs.
8. **PLT Tools** will create graphics and summary files (CSV format) based on the merged files. Timestamp for the outputs will be the concatenation of the timestamps of individual runs.

Combining Likelihood Profiles

On occasion, the boundaries that a user has selected for a likelihood profile are incorrect and the user wishes to extend the profile in one or both directions. This could be accomplished by repeating the entire process with broader limits. However, **PLT Tools** allows the user to piece together the results of two or more analyses. This utility will be explained by means of an example:

1. The user selected a lower limit of 100 and an upper limit of 200 for a likelihood profile. The graphic of the likelihood profile suggested that a higher upper limit was appropriate.
2. In Project Controller, select new limits for the likelihood profile, *e.g.*, a lower limit of 220 and an upper limit of 300. Select a reasonable value for # of Intervals. Click **NONMEM** or **NONMEM + Graphics**.
3. When this new analysis completes, **PLT Tools** will have access to two sets of likelihood profiles.
4. In the `WorkingFolder`, create a new text file (extension must be either `.txt`, `.ctl`, or `.con`) with this exact syntax:
 - a. The first line must be:
`PLTTOOLS: LIKELIHOOD GRAPHICS`
 - c. The second line must be:
`THETA N`
where N is an integer indicating the parameter for which the likelihood profile is being performed.
 - c. On subsequent lines, enter the 13-character timestamps for each of the likelihood profiles. A minimum of one entry is required.
 - d. The user can select (sequential) subsets from each of the likelihood profile analyses. For example, if an analysis was based on 10 intervals (resulting in 12 runs [one in which all parameters were estimated; one at the beginning of the first interval; and one at the end of each of the 10 intervals]), the user could select a subset (in this case, omitting the run at the

beginning of the first interval) with the following code:

```
110706-161739 3 12
```

where the first part of the text is the timestamp, the second part is the starting position, and the third part is the ending position. Non-contiguous runs can be selected by repeating the timestamp:

```
110706-161739 2 5
```

```
110706-161739 7 9
```

Note that run #1 (for which the THETA is estimated, not fixed) is always included in the analysis regardless of whether it is identified as described above.

Portions of text are separated by a single space.

5. Select this text file as the Control Stream. Select the Graphic Scripts that was used for the bootstrap analyses.

6. Click **NONMEM** or **NONMEM + Graphics**.. **PLT Tools** will bypass most of its usual procedures. Instead, it attempts to do the following:

- a. read the text file containing the timestamps
- b. find the likelihood profile parameter outputs associated with those timestamps
- c. merge these files.

7. **PLT Tools** will create graphics and summary files (CSV format) based on the merged files. Timestamp for the outputs will be the concatenation of the timestamps of individual runs.

Automation

PLT Tools can conduct a series of runs in an automated manner. Potential uses for this include:

1. A user has conducted a series of analyses and needs to revise the graphics. For example, a covariate was omitted, new covariate data is available, or a label was entered incorrectly.
2. New data are available (*e.g.*, additional subjects). The user expects the analysis pathway to be similar and wishes to re-run a set of analyses.

In Windows, automation may not work if **PLT Tools** is installed in a location other than Program Files (or Program Files (x86)). This results from Windows not recognizing an .exe file as an executable if it is not located in the expected location. If this prevents use of automation, contact support@PLTsoft.com for suggestions.

Note: Automation applies only to "Normal" NONMEM runs, not to covariate searches, visual predictive checks, bootstrap, likelihood profiles, or jackknife.

Note that automation applies only to "Normal" NONMEM runs, not to covariate searches, visual predictive checks, bootstrap, likelihood profiles, or jackknife.

Note that automation applies only to “Normal” NONMEM runs, not to covariate searches, visual predictive checks, bootstrap, likelihood profiles, or jackknife.

Two different approaches are available to use automation in **PLT Tools**. For both, an automation script is necessary. In the first approach, the user prepares the automation script and has complete control over its contents. In the second, **PLT Tools** creates the automation script using one of four different approaches.

1. *User prepares an automation script:* The user can create a file with the extension .plta (PLT Tools automation); the location of the file is not important. The file contains a series of instructions to **PLT Tools**. A sample file appears in this folder. Once this file is created, clicking on the file should start **PLT Tools** in an automated manner.* A sample automation session is displayed in **Figure 19**.

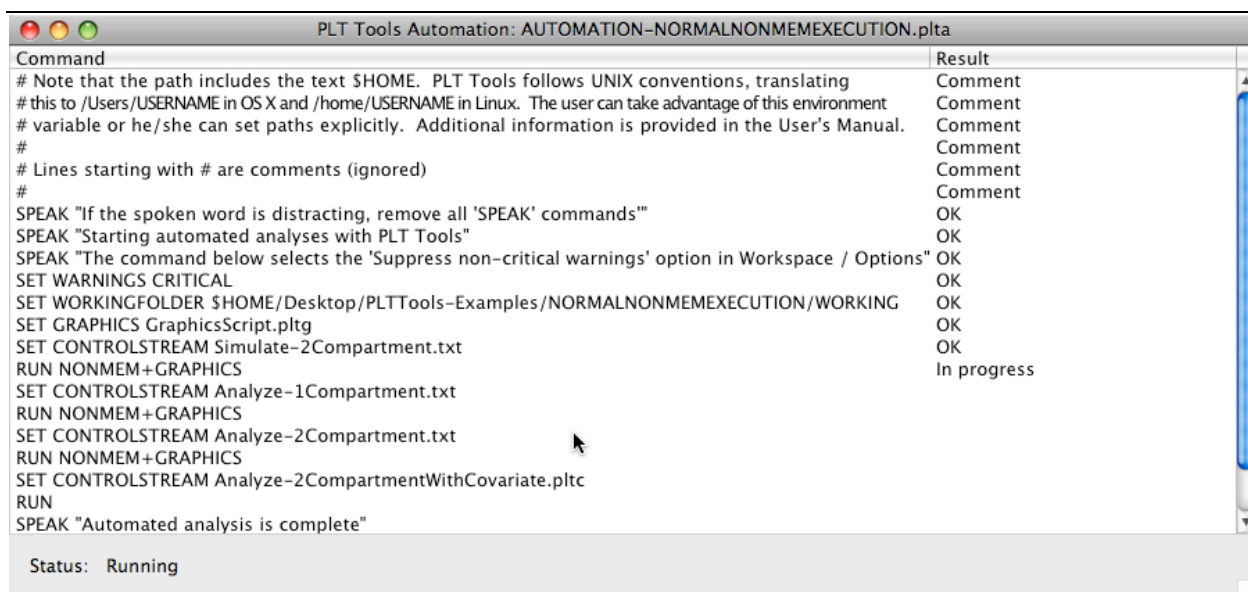


Figure 19. A sample automation session is displayed.

The left portion shows the commands contained in the automation script (blank lines in the script are omitted). The right portion shows the response to these commands.

* * At the first use, it is necessary to inform the operating system to “associate” plta files with **PLT Tools** (this is required only once).

Windows: right-click, select “Open With”, then designate PLTTools.exe. Check “Always use the selected program to open this kind of file”. If the PLTTools folder is not located in “Program Files” (or “Program Files (x86)”, Windows may not allow the user to associate a plta file with **PLT Tools**. Contact support@PLTsoft for recommendations as to how to overcome this restriction.

OS X: right-click (option-click), select “Open With”, then designate PLTTools.app. Check “Always Open With”.

The script consists of the following elements (in their expected order):

`SET WORKINGFOLDER /PATH/TO/WORKINGFOLDER` (required). The full path must be provided. In Windows, if the drive letter is omitted, **PLT Tools** assumes that the file is on the C: drive.

`SET GRAPHICS NAMEOFGRAPHICSSCRIPT` (optional; if not provided, **PLT Tools** attempts to auto-select a script – if not successful, default graphics are created)

`SET CONTROLSTREAM NAMEOFCONTROLSTREAM` (optional if `RUN GRAPHICS ONLY` is selected – see next command)

`RUN NONMEM+GRAPHICS` or
 `RUN NONMEM ONLY` or
 `RUN GRAPHICS ONLY`

The following commands are optional. They can appear in any position:

`SET WARNINGS ALL` or
 `SET WARNINGS CRITICAL:`

Certain messages from **PLT Tools** appear in popup boxes. These messages prevent **PLT Tools** from moving to the next analysis during automation. If `SET WARNINGS CRITICAL` is selected, only “Error” messages are reported (“Information” and “Warning Messages” are not reported). However, a “Special Message” appears after the first run to advise the user that messages are being suppressed. If the command is omitted, all messages appear.

`SPEAK <text>` speak the given text with speech synthesis, if available.
`PRINT <text>` display the given text in a message (alert) box.
`QUIT` close **PLT Tools** when complete

Additional features:

1. `WORKINGFOLDER` needs to be set only once. Subsequent `SET WORKINGFOLDER` commands can be used to declare the same or different `WORKINGFOLDER`.
2. Same for `GRAPHICS`
4. If there is only one Control Stream available, **PLT Tools** will auto-select that Control Stream.
3. If only one graphics script is available and there is no `SET GRAPHICS` command, **PLT Tools** will auto-select that graphics script. To prevent this, use the command `SET GRAPHICS ""`
4. `RUN` without an option executes `RUN NONMEM+GRAPHICS`

Environment variables: Certain environment variables are understood by the Automation tool in **PLT Tools** and can be used to simplify entering a `/path/to/file`. These environment variables differ by platform:

Windows:

`%HOMEDRIVE%` typically maps to C:

`%HOMEPATH%` differs between versions of Windows:

XP: maps to \Documents and Settings\USERNAME*
Vista: maps to \Users\USERNAME*
OS X: \$HOME maps to /Users/USERNAME*

Use of these environment variables is illustrated in the examples.

Two sample automation scripts are provided in the PLTToolsExamples folder. The user is encouraged to use these as templates to develop their own scripts.

2. PLT Tools generates the automation script: The pull-down menu for Control Stream in the Project Controller tab (**Figure 20**) includes five entries at the bottom:

AUTOMATION (based on):

- All files in PROJECTFOLDER/TEXTFILES/CONTROL
- All timestamps in Run Listing
- All timestamps in file WORKINGFOLDER/LocalList.txt
- All Control Streams (.txt, .text, .ctl, .con, .pltc) in WORKINGFOLDER.

The final four of these are options for selecting Control Streams to be run in an automation process. If the user selects any of these four, **PLT Tools** assembles the automation script based on the appropriate set of Control Streams. The automation script is then called by **PLT Tools** and each of the Control Streams is run in sequence. If the user is configured for parallel runs (which requires a license), **PLT Tools** creates multiple scripts and attempts to run these scripts in parallel.

Automation scripts can take advantage of either “NONMEM + Graphics”, “NONMEM Only”, or “Graphics Only”. If the intent is to update graphics, the “Graphics Only” option can be selected.

Changing the Dataset for Automation Runs: Under some circumstances, the data may have changed and the name of the dataset differs from that in the original Control Stream. **PLT Tools** can be instructed to use a new dataset by creating a text file:

AUTOMATION-ReplacementDataset.txt

in the WORKINGFOLDER. This file contains the absolute or relative path to the dataset, referenced to WORKINGFOLDER. For example, if the dataset is in the DATAFILES folder, the path might be:

../DATAFILES/NewDataset.csv

or:

C:\PATH\TO\PROJECTFOLDER\DATAFILES\NewDataset.csv

Graphics Scripts in Automation: If the user has prepared a Graphics Script, **PLT Tools** will attempt to use the auto-select procedure to determine which Graphics Script to use. However, if there is more than one Graphics Script in the Scripts-Graphics folder, **PLT Tools** will not be able to make the selection and will prepare default graphics instead. This can be overcome by assuring that there is only one .pltg file in the Scripts-Graphics folder (others can be placed in a sub-folder, *e.g.*, UNUSED.GRAPHICS.SCRIPTS).

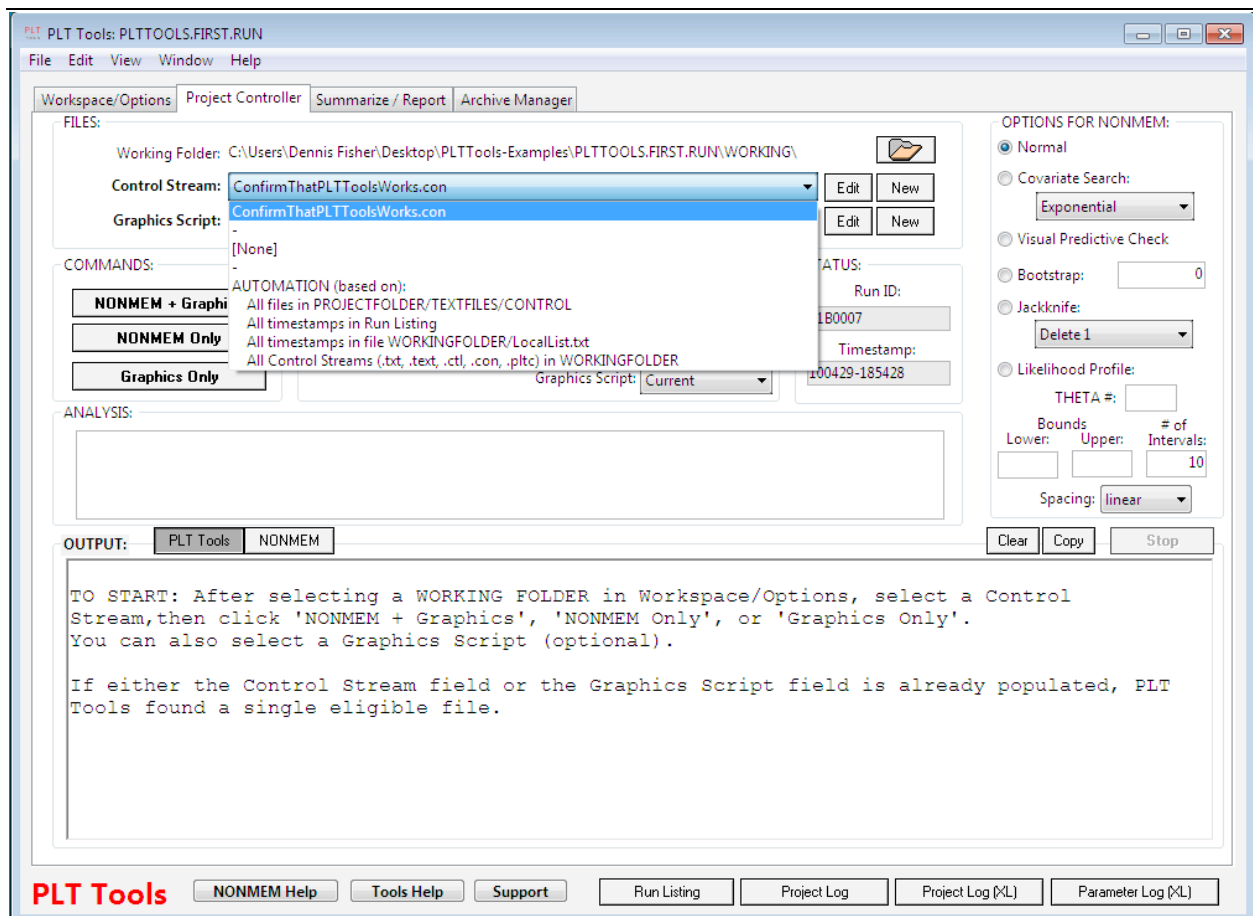


Figure 20. The pull-down menu displays choices for automation.

Command Line Execution

PLT Tools can be called from the command line. Relevant files - PLTTools-CommandLine.bat (OS X: PLTTools-CommandLine) and CommandLine.R - exist in the PLTTools-Support folder. To run **PLT Tools** from the command line, the user needs to do the following:

1. Open a Command Prompt (OS X: terminal) window
2. Navigate to a WorkingFolder

3. Type:

Windows: \path\to\PLTTools-Support\PLTTools-CommandLine CONTROLNAME

OS X: /path/to/PLTTools-Support/PLTTools-CommandLine CONTROLNAME

4. **PLT Tools** will create an automation script, then execute that automation script.

5. Additional arguments that can be provided to the command line are the name of a Graphics Script and/or commands to execute NONMEM Only (“NO” or “NONMEMONLY”) or Graphics Only (“GO” or “GRAPHICSONLY”). For example, to execute NONMEM only, the command might be:

Windows: \path\to\PLTTools-Support\PLTTools-CommandLine CONTROLNAME NO

OS X: /path/to/PLTTools-Support/PLTTools-CommandLine CONTROLNAME NO

The order of arguments is not important.

6. If the user adds /path/to/PLTTools-Support to the path (as an environment variable), the command can be shortened to:

PLTTools-CommandLine CONTROLNAME

The files PLTTools-CommandLine.bat (or PLTTools-CommandLine) and CommandLine.R might require editing to address local issues. Contact support@PLTsoft.com for assistance.

Project Log, Parameter Log

After each NONMEM run, certain information is tabulated into these two logs. The logs record information for at the level of the PROJECTFOLDER; thus, if the user has two or more WORKINGFOLDERS, information from NONMEM runs in all of these folders will appear in the logs.

Project Log

This log contains the following information: timestamp, the approach used in the analysis (Normal, VPC, Bootstrap, Likelihood Profile, Jackknife), the name of the Control Stream, and text that appears in the \$PROBLEM record in the Control Stream, whether simulation or estimation (or both) was employed, the number of the ADVAN used, FO vs. FOCE, the number of function evaluations, the number of significant digits, whether the covariate step was requested and/or completed successfully, the objective function, and the number of THETAS, OMEGAS, and SIGMAS in the model. Selecting “Save As” from the “File” menu produces a comma-delimited file named after the PROJECTFOLDER.

If the user selects a line in the Project Log, buttons at the bottom of the window become active. These buttons permit the user to view the Graphics file, the Brief Summary, or the Results or Summary files for that run. In addition, if the user wants to re-use an archives Control Stream, click the “Copy” button on that row: the control stream will be copied to the Working Folder and the Control Stream field will be populated with the correct entry.

Project Log (XL)

The Project Log can also be opened as a CSV (comma-separate values) file *via* Microsoft Excel or Open Office.

Parameter Log

This log contains the following timestamp and the values for THETA (up to 40), OMEGA (up to 20) and SIGMA (up to 10) in the model. The log opens as a CSV file, typically using Excel. Selecting “Save As” from the “File” menu produces a comma-delimited file named after the PROJECTFOLDER.

Using the Graphics Editor

Conduct of a NONMEM analysis depends on the availability of high-quality graphics. Using the Graphics Editor, the user is able to create consistent high-quality graphics with minimal effort and with no knowledge of the R engine with which the graphics are created. Three sets of graphics are created, one relevant to each NONMEM run, the second displaying each pair of covariates, the third displaying observations (“data checkout graphics”).

Graphics Script

PLT Tools creates high-resolution graphics, saved in the PDF format, using R. The user does not need to have any knowledge of these graphics packages. Instead, the Graphics Editor is used to create scripts called by R. Use of the graphics editor is explained below.

Graphics for each NONMEM run:

These graphics can be best understood by examining a sample graphic provided in the installation download (see [Examples-FullyPopulated/Graphics](#)). The graphics fall into two categories:

1. Graphics related to observations. These include:

a. **By-subject plots:** For each subject, a panel displays observations (points), population predictions (blue line), and *post hoc* predictions (red line) vs. time. If a limit of quantification (LOQ) is identified in the Graphics Editor, the LOQ is displayed (dotted line), one or more BQL values are displayed with “X”, BQL values used in the analysis (*i.e.*, those with EVID=0) are displayed as green circles, and values > LOQ not used in the analysis are displayed as red circles.

If the “Multiple Periods” option is selected (and appropriate information provided), each period containing ≥ 1 observation is displayed in a separate panel; in addition, a header indicates the period. This option is useful for those situations in which samples are obtained over a lengthy period but the critical period during which concentrations are changing is short, *e.g.*, a food-effect study with two doses separated by a lengthy interval.

Suppressing certain periods: Consider a multi-dose study with daily trough samples followed by intensive sampling on Day 7. Enter -1 for all samples before Day 7; then enter a positive value on Day 7 (e.g., 1 or 7). When **PLT Tools** creates graphics for specific periods, it will ignore periods with values < 0; i.e., a graphic will be prepared for the Day 7 values only.

If the value for **TIME** on the first record for a subject is larger than 0 (e.g., if times are referenced to midnight preceding the first dose), **PLT Tools** can re-zero the times referenced to the time of the first record for that subject. This is accomplished by selecting the “Re-Zero Times” option in the By-Subjects tab of the Graphics Script.

An arbitrary upper limit for the x-axis can be imposed by entering an x-axis upper limit in the Graphics Script. If this option is invoked, times are re-zeroed (as in the previous paragraph) so that the time upper limit is referenced relative to the first record for that subject.

If provided in tables and a Graphics Script is populated with the appropriate information, certain demographic information (age, gender, race, weight, dose group, e.g., mg/kg) and dose magnitude (optional) are displayed.

Doses are displayed as arrows or horizontal lines, depending on the data in the **AllRecords** file. If **RATE** does not exist (e.g., because all doses are bolus), doses are displayed as arrows. If **RATE** does exist (and there is at least one value > 0), horizontal lines (with vertical bars at each end) mark the start and end of the infusion. The end of each infusion is determined as the sum of the start time and **AMT/RATE**. If the infusion is brief (less than 5% of the time scale of the panel), **AMT** appears *below* the line; if the infusion is long, **RATE** appears *above* the line. The user can select whether to display the magnitude of bolus doses at the first dose only, all doses, or neither.

If the user has indicated that there are > 1 dependent variables (“Multiple DV”), there are separate sets of pages for each dependent variable (with headers on each page indicating which DV is displayed).

Vertical lines can be displayed in by-subjects panels – these lines might be useful for marking events that occur during the conduct of a clinical trial.* To display vertical lines, the **AllRecords** table must contain a data item named **VLCn** (where n has values 1, 2, 3, 4, or 5). **PLT Tools** evaluates whether a column with that name exists and that the data for that subject contain only a single non-negative value. If so, a vertical line is displayed at that x-value. Colors, corresponding to integer values 1-5 are green, red, blue, orange, magenta, black.

If any records contain samples for which **DV** > 0 and either **MDV** = 1 or **EVID** = 2 (these would be records excluded from the analysis, possibly because the concentration was reported as BQL), an additional graphic is created showing population and *post hoc*

* Examples include change in a therapeutic regimen or occurrence of antibodies.

predictions for these samples vs. time-after-dose (calculated using both explicit and implicit [ADDI] doses). This graphic might be valuable if initial screening of BQL values excluded all such values from the analysis and the user wanted to learn whether these samples were predicted to be < LOQ.

DISPLAYING THE TIME COURSE SURROUNDING A SAMPLE: If the NONMEM dataset has been created with **PLT Data**, a column name NSAMP is automatically added to the dataset. The column contains zeros for all records except samples, then sequential integers for each sample record (starting at 1). Alternatively, the user might have added such a column to the dataset.

A column with that name can be used to create graphics that surround the sample. This is accomplished by entering "NSAMP | X.X" (no spaces or quotes) in the Within-Subject column in the By-Subject Graphics tab in the Graphics Editor. **PLT Tools** will then create a temporary column in the dataset that extends the sample number entries to all records for X.X time units preceding the time of the sample. For example, if a sample were obtained at 12.7 days and X.X were 2.0, all records with TIME ≥ 10.7 and TIME ≤ 12.7 would be displayed in a separate panel.

If " | X.X" is omitted, only the observation and PRED (and IPRED) at the time of the observation will be displayed.

b. **Spaghetti plots:** Each page contains two panels, the left panel based on the population fit, the right panel based on the *post hoc* fit. For each panel, the x-axis is time; the y-axis is the ratio (or difference [user-selected]) of observed and predicted values. For each subject, a thin blue line connects all values. If a limit of quantification has been declared, red dots indicate ratios (or differences) for these observations. A smoother (Supersmoother®) is displayed with a thick line; a green line does not include BQL values, a red line includes BQL values. Two statistics are displayed above each panel, the median prediction error and the median absolute prediction error.

DISPLAYING THE TIME COURSE SURROUNDING A SAMPLE: See text box above.

c. **Observed vs. Predicted:** A series of panels displays observed vs. predicted values on linear-linear or log-log scales. If LOQ has been provided, it is displayed and the axes of log-log displays are adjusted accordingly; if LOQ has not been provided, log-log scales are adjusted to the data. Linear scales use zero as the lower limit and LOQ, if provided, is displayed. For some linear-linear displays, the axis upper limit is trimmed so as to magnify data at the low end; this truncates some data from the high end (the fraction of data truncated is indicated in the graphic). A smoother (Supersmoother®) appears in all panels).

d. **Other goodness of fit graphics:** A series of panels display population or *post hoc* residuals vs. either time or the corresponding (population or *post hoc*) predictions. If *post hoc* residuals (IRES) have not been provided in a table by the user, they are calculated during post-processing. *Post hoc* weighted residuals are not displayed unless provided. Graphics of weighted residuals are scaled identically; in order to ensure that extreme values

are not omitted, values > 4 or < -4 are displayed at ± 4.5 . A smoother (Supersmoother[®]) appears in all panels; for weight residuals, it is based on untransformed values. If the user has adapted the Control Stream to permit calculation of conditional weighted residuals, these are displayed.

2. Graphics related to subjects.

a. **Covariate plots:** Three sets of covariate plots are provided. For each set, the x values are the covariates entered by the user in the Graphics Editor; in addition, index plots (x-axis is ID) are displayed. If a covariate contains only a single unique value, it is not displayed. The y-axis differs between the three sets of plots. The first set displays *post hoc etas*. The second set displays the ratio of the *post hoc* value to the population value ($\exp(\eta)$); these graphics are limited to the set of *omegas* identified as “exponential” in the Graphics Editor. The third set displays *post hoc* values on the y-axis. For all three sets, if the set of y values is unique, it is not displayed.

For covariates identified as continuous (age, weight, height and ID are assumed to be continuous), a smoother (Supersmoother[®]) and a linear regression (with the corresponding *r* and *P* values) are displayed. In addition, the range of fitted y values is displayed. For a covariate identified as categorical, *t* tests compare y values for each x to those for the remaining x values. Finally, a line marks the median y value.

For the first and second set, the y-axis is scaled automatically with one exception: the minimal range for the first set is -0.4 to 0.4, the minimal range for the second set is 0.5 to 2.0. Finally, *post hoc eta* values reported as 0 are omitted from all displays and all calculations, the assumption being that values of zero imply that an *eta* value was not calculated for that subject (for example, in a crossover study to determine absolute oral bioavailability), a subject who received only intravenous doses would have a value of zero for the *eta* for bioavailability).

b. **Eta vs. Eta plots:** For each pair of *omega* values, *post hoc etas* are displayed against each other. A smoother (Supersmoother[®]) and a linear regression (with the corresponding values for *r* and *P*) are displayed. In addition, a line marks the median value of each of the sets of *eta* values.

Data Source for Graphics

Internally, graphics obtain data from three sources, a file called `AllRecords`, a file called `FirstRecords`, and a covariate file. The `AllRecords` file contains the same number of rows as the input data and is generated by NONMEM's table step (see “Modifications needed for the NONMEM Control Stream”); this file MUST be requested by the user if graphics are to be created. This file should contain all information relevant to the observations and predictions, e.g., subject ID, time, observed values, population predictions, and *post hoc* predictions. If the file is not created, the user is informed that graphics cannot be created.

The `FirstRecords` file contains the same number of rows as the number of subjects identified by NONMEM. There are two means by which the `FirstRecords` file can be

created. The preferred approach is for the user to create a second table using a \$TABLE record in the Control Stream; the user should select the FIRSTONLY option (this creates a table with a single record per subject). This file should contain all information relevant to the subjects, *e.g.*, subject ID, *post hoc* parameter values, and *post hoc eta* values. Alternatively, the user can output these values to the FirstRecords files described above. If a FirstRecords file is created by NONMEM, it is used to create graphics. If this file is not created by NONMEM, the AllRecords file is searched to determine availability of *post hoc etas* and *post hoc* parameter values. If these values are identified in the AllRecords file, a FirstRecords file is created from the appropriate rows (the first row for each subject) and the appropriate columns (ID plus parameters and *etas* listed in the Graphics Editor); this file is saved in the folder TABLES/FIRSTRECORDS-PROCESSED.

Covariate data is obtained from one of two sources. If the user has selected “Use Covariates File” in the Graphics Editor, the file identified using the **Browse** button is the source. If the user has not selected “Use Covariate File”, AllRecords is the source: as with FirstRecords, the AllRecords file is searched for the appropriate covariates. If these values are identified in AllRecords, a covariates file is created from the appropriate row (the first row for each subject) and the appropriate columns (ID plus covariates identified in the Graphics Editor); this file is saved in the folder TABLES/DEMOGRAPHICS.

The following rules apply to the Covariate file:

1. There should be a single row per subject (order not important). The covariate file can contain subjects not contained in the input dataset. For example, during assembly of a dataset, covariate data may be available for 50 individuals but plasma concentration values may be available for only 20 of these subjects. The input dataset would include only the 20 subjects whereas the Covariate file could contain all 50 (unique IDs required).
2. Headers in the Covariate file must appear in the first row and there should be no empty rows. Spaces should be avoided.
3. The file can be delimited either with tabs, commas, or spaces. The user must indicate the delimiter in the Graphics Editor. It is important that the character used as the delimiter is not part of any of the entries, *e.g.*, if a comma is used as the delimiter, ensure that none of the text entries contain commas.
4. All columns that are to be used as covariates must be entered as numbers only. Missing values should be entered as real numbers < 0 (*e.g.*, -1 or -99). Covariate graphics highlight values < 0 in red and exclude these values from statistics.

Users familiar with the restrictions in NONMEM tables will readily appreciate the opportunities provided by the use of the covariate file. Specifically, early NONMEM versions restrict the number of columns of data that can be read as input (19 + MDV) and written to a table. Consider a situation in which the analyst wishes to screen 20 covariates. It is not possible to read all these covariates into NONMEM, then, write them to the FirstRecords file. An inefficient approach would be to re-run NONMEM with different datasets, each of which contained different sets of covariates. A more efficient approach (recommended) is to create a

single dataset containing all covariates but add “=DROP” to all covariates not required in the analysis; this permits the user to maintain a single dataset. In this situation, tables output by NONMEM do not contain the covariates; however, **PLT Tools** can obtain the appropriate covariates from the covariates file.

NOTE: All covariate graphics are based on the values provided in the covariates table or, if this table is not provided, on the first value identified for each subject in the AllRecords file. Thus, covariate graphics may be flawed if a covariate value changes with time.

NOTE: **PLT Tools** depends on the AllRecords file (using the first value identified for each subject) to determine median covariate values. Therefore, it is imperative that the user not use null values (“.”) for covariates, despite this being allowed in NONMEM.

Recommendations for Source Files

Unless the NONMEM problem is extremely small (few parameters, few *omegas*, few covariates), the optimal approach is for user to create a Covariate file (*e.g.*, at the same time that the dataset is created) and to generate both AllRecords and FirstRecords tables in NONMEM using the recommendations listed above.

The Covariates file is not needed (although still acceptable) if the following conditions are met:

- a. all the covariates can be read into NONMEM *via* \$INPUT
- b. all the covariates can be output via \$TABLE to AllRecords.

The FirstRecords file is not needed (although still acceptable) if the following condition is met: all the *post hoc* parameters and *post hoc etas* can be output via \$TABLE to AllRecords.

Truncated IDs

When NONMEM creates tables, the default is to truncate all values to 5 significant digits in scientific format. This contrasts to NMTRAN's ability to read 12 digits during input. This may lead to problems. For example, if two subjects have IDs 100001 and 100002, both will appear in the output tables with ID 1.0000E+05. **PLT Tools** attempts to identify this issue by comparing the number of unique ID's in the NONMEM tables to the number of subjects identified by NONMEM (TOT. NO. OF INDIVIDUALS:). If these differ, the following happens:

1. A message is sent to the Output window. This same message appears later in a pop-up dialog. The message explains the discrepancy and the solution.
2. **PLT Tools** finds the correct IDs in the FDATA file (where they appear with all digits) and replaces the values in AllRecords and FirstRecords objects read by **PLT Tools** (but not in the stored files).
3. Having corrected the IDs, **PLT Tools** is able to post-process, including creating graphics.

The user has three options:

1. Allow **PLT Tools** to resolve the problem. The warning message will appear during each run.

2. Create a new ID that contains ≤ 5 significant digits. If this approach is used, the user can still obtain the original ID in By-Subject graphics. To accomplish this, the user must have a Covariates file. In the Covariates file, the original ID is labeled "IDforGRAPHICS". The new ID is labeled with the "Name for ID in Covariates File". When **PLT Tools** creates graphics, it will match the corrected ID column in AllRecords (and FirstRecords, if it exists). BySubject graphics will contain both the new ID and the IDforGRAPHICS values.
3. Increase the number of digits in tables created by NONMEM. Instructions are available by clicking NONMEM help in **PLT Tools**, then entering \$TABLE (see the FORMAT section). An option that is usually successful for the \$TABLE record is

```
FORMAT=s1PG13.6
```

Effect of Truncated IDs on XPOSE graphics: If the user creates XPOSE graphics from **PLT Tools** and NONMEM tables contain truncated (ambiguous) IDs, certain XPOSE graphics (*i.e.*, anything at the 'subject' level) will be flawed. To prevent this from happening, **PLT Tools** will identify NONMEM runs in which this issue occurred and will not permit creation of graphics from those runs. This change was implemented in Version 5.2.0.

Updating Covariates

In certain instances, the user wishes to update graphics with new covariate data. For example, at the time that an analysis is performed, certain laboratory or demographic data may not be available. If graphics were created using a Covariates file, update that file. Then, in the "Covariates" field in the **Graphics Only** area of the Project Controller, select "Updated". When **Graphics Only** is selected, **PLT Tools** will obtain covariate data from the Covariates file.

Graphics based on covariate data from a revised Covariates file may misrepresent the relationship between parameters and covariates. For example, if clearance varied with weight and if the weight entered into the original analysis was flawed, revising the value for weight in these graphics will not revise the flawed analysis performed by NONMEM. Thus, graphics created from revised Covariates files are labeled: "Covariates obtained from CovariateFileName. These covariates may differ from those used by NONMEM".

Populating the Graphics Editor

Completing the necessary entries in the Graphics Editor requires 5-10 minutes at initiation of a project; the graphics script can be re-edited at a later date and graphics for previous runs can be re-created. Many of the fields contain default values that do not require changes. When **PLT Tools** reads the output of the Graphics Editor, the user is advised that certain default values have been applied. If the user determines that these defaults are not optimal, change the entries in the appropriate fields, save the graphics scripts, and re-create the graphics. The remainder of this section how each field of the Graphics Editor can be entered.

Tab 1: THETAS, ETAS, SIGMAS:

This tab (**Figure 21**) identifies the names of the elements that will be used in the graphics and tables.

GraphicsScript.pltg

THETAS, ETAS, SIGMAS | Covariates | By-Subject Graphics | Spaghetti Graphics | General Layout/Options | Text Mapping

☒ Calculate Half Lives

Compartments: 2

THETA # is optional. If provided, 'Label in Graphics' is displayed in the "real-time" update window.
 * LINK: If a number is entered in this row, the 'Label in Graphics' from THETAS is used (no need to enter 'Label in Graphics').
 To re-order rows, drag a non-selected row. Use the PLUS and MINUS buttons to add or delete rows.

THETAS				
#	NONMEM Term	CL/V/ka	Label in Graphics	Search
1	CL	CL1	Clearance/F	<input checked="" type="checkbox"/>
2	V1	V1	V1/F	<input type="checkbox"/>
3	CLRA	CL2	Distributional Clearance/F	<input type="checkbox"/>
4	V2	V2	V2/F	<input type="checkbox"/>
5	KA	-	Absorption Rate	<input type="checkbox"/>
6	LAG	-	Absorption Lag	<input type="checkbox"/>

ETAS			
#	Label in Graphics	Exp	LINK*
1		<input checked="" type="checkbox"/>	1
2		<input checked="" type="checkbox"/>	2
3		<input checked="" type="checkbox"/>	3
4		<input checked="" type="checkbox"/>	4
5		<input checked="" type="checkbox"/>	5
6		<input checked="" type="checkbox"/>	6

SIGMAS	
#	Label in Tables
1	Proportional Error
2	Additive Error

PLT Tools

☐ View real-time contents of Editor Output

Save

Figure 21. Tab 1 of the Graphics Editor.

Half-lives: If a linear 1-, 2-, or 3-compartment model is described in terms of clearance, distribution clearance(s), and volume(s) of distribution, **PLT Tools** can calculate distribution and elimination half-lives, steady state distribution volume, and fractional coefficients. To accomplish this, check the box "Calculate half-lives" and select the appropriate number of compartments using the pull-down menu.

THETAS: If the user outputs *post hoc* parameters to the FirstRecords (or AllRecords) file, these parameters are identified here.

Column 1 (#): This column is optional. If parameters correspond exactly to THETA terms, then it is recommended that you enter the corresponding numbers here. For example, in the following code:

```
CL = WT * THETA(1) * EXP(ETA(1))
```

entering the value 1 for CL is appropriate. However, with the following code:

```
CL = (THETA(1) + THETA(2) * CRCL) * EXP(ETA(1))
```

CL does not correspond directly to either THETA(1) or THETA(2).



Column 2 (NONMEM Term): This column contains the term used in the NONMEM analysis, e.g., CL. These terms are case-sensitive, i.e., if the term CL is used in NONMEM, the term CL will not process correctly. During the creation of tables, NONMEM truncates all names to four characters; the user is responsible for ambiguities created in this situation (e.g., if the user outputs two parameters TEMP1 and TEMP2, these both appear in a table as TEMP).

Although the user can enter a term longer than four characters, longer entries in the left column are truncated to four characters.

Column 3: This column indicates whether these terms refer to volumes and/or clearances for linear pharmacokinetic models; this is necessary if the user requests that half-lives of a compartmental model be calculated (see Half-Lives above). Consider the following situation: NONMEM terms for the four structural parameters of a two-compartment model are CL, CLDIST, V1 and V2. These are designated CL1, CL2, V1, and V2, respectively, in the second column. Having designated the four terms associated with a two-compartment model, **PLT Tools** is able to generate half-lives for that model. If the user has not selected the correct terms for the designated model (e.g., the user designates a 3-compartment model but selects only CL1 and V1), warnings appear in a separate window. In addition, if an absorption rate constant is included in the model, absorption half-life will be calculated if the user assigns "ka" to a NONMEM term,

Column 4 (CL/V/ka): This column indicates the name that will be used in the graphics, e.g., "Clearance (L/h)" might be appropriate for CL. Symbols (e.g., Greek characters) are not permitted.

Column 5 (Search): This column indicates whether a THETA should be included in a Covariate Search.

When a new script is being populated, a single row is available for entry. Rows can be added or deleted using the  and  buttons at the upper right.

ETAS: If the user outputs *post hoc etas* to the FirstRecords (or AllRecords) file, these terms are identified here

Column 1 (#): This column contains the numeric value (integers ≤ 99) of the *eta* term.

Column 2 (Label in Graphics): This column indicates the text describing the parameter to which the *eta* term is applied. In graphics, this text is pre-pended with the text "ETA n" where "n" is the integer associated with the *eta*. The same rules apply to this column as to the third column of the Parameters field. If a

Column 3 (Exp): This column permits the user to designate whether the user has invoked EXP(ETA(n)) vs. 1 + ETA(n) in NONMEM. In that certain displays involve exponentiation of *eta* terms, only those terms for which this option is checked will be exponentiated.

Column 4 (LINK): This column permits the "link" an ETA term to a THETA term. For example, the code:

```
CL = WT * THETA(1) * EXP(ETA(1))
```

associates ETA(1) to THETA(1). If ETA(1) is not applied to any other THETA, then THETA(1) and ETA(1) are "linked". If so, the user should enter the THETA # corresponding to the ETA # in the LINK column. This accomplishes two purposes:



1. The "Label in Graphics" for the ETA need not be re-entered – the label for the corresponding THETA will be applied.

2. The table summarizing population parameters will associate the inter-individual variability associated with a linked ETA to the corresponding THETA. For example, if the Control Stream contained the code:

$$CL = WT * THETA(1) * EXP(ETA(1))$$

and the Graphics Script provided the label "Clearance" for CL1, the table would present all of the following in a single entry:

- the term "CL"
- the associated label "Clearance"
- the typical value for THETA(1)
- the associated interindividual variability

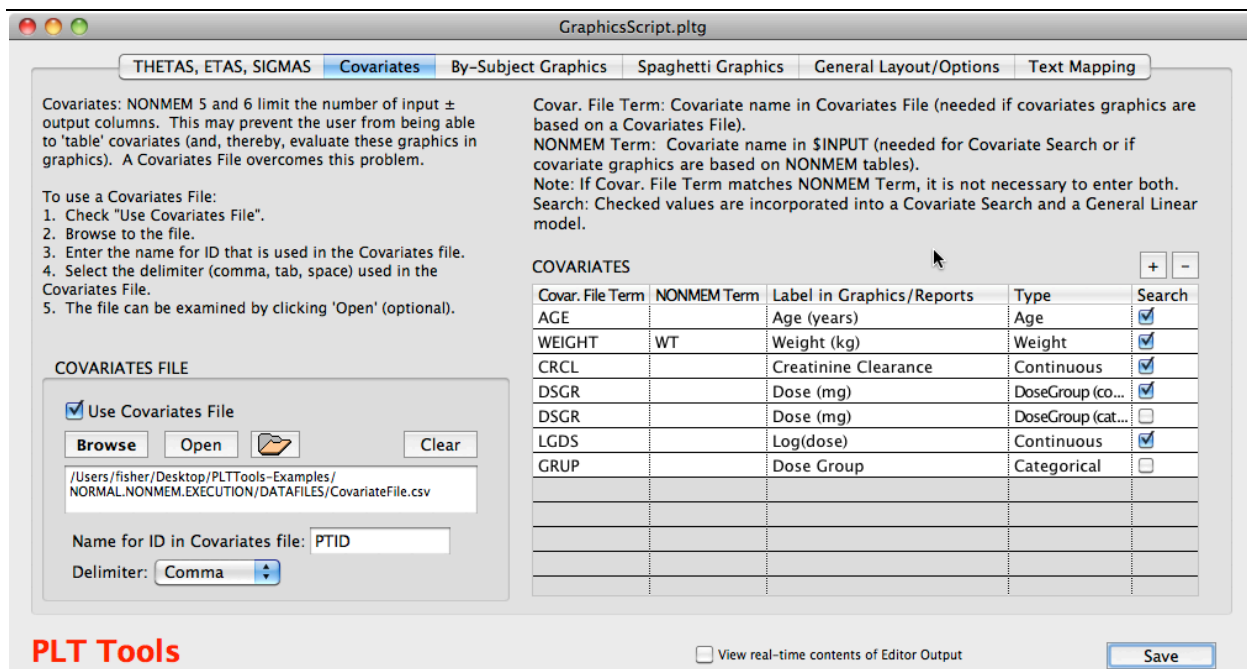
When a new script is being populated, a single row is available for entry. Rows can be added or deleted using the  and  buttons at the upper right.

SIGMAS: Entries here are used in the Brief Summary and in Population Parameter tables.

If there are > 1 SIGMA terms in a model (e.g., both additive and proportional error or different error terms for parent/metabolite, different studies, or PK/PD components of the model), entering names

Tab 2: Covariates

This tab (**Figure 22**) identifies covariates to be examined in covariate graphics and in a Covariate Search.



GraphicsScript.pltg

THETAS, ETAS, SIGMAS **Covariates** By-Subject Graphics Spaghetti Graphics General Layout/Options Text Mapping

Covariates: NONMEM 5 and 6 limit the number of input ± output columns. This may prevent the user from being able to 'table' covariates (and, thereby, evaluate these graphics in graphics). A Covariates File overcomes this problem.

To use a Covariates File:

1. Check "Use Covariates File".
2. Browse to the file.
3. Enter the name for ID that is used in the Covariates file.
4. Select the delimiter (comma, tab, space) used in the Covariates File.
5. The file can be examined by clicking 'Open' (optional).

COVARIATES FILE

☒ Use Covariates File

/Users/fincher/Desktop/PLTTools-Examples/
NORMAL.NONMEM.EXECUTION/DATAFILES/CovariateFile.csv

Name for ID in Covariates file: PTID

Delimiter:

Covar. File Term: Covariate name in Covariates File (needed if covariates graphics are based on a Covariates File).

NONMEM Term: Covariate name in \$INPUT (needed for Covariate Search or if covariate graphics are based on NONMEM tables).

Note: If Covar. File Term matches NONMEM Term, it is not necessary to enter both.

Search: Checked values are incorporated into a Covariate Search and a General Linear model.

Covar. File Term	NONMEM Term	Label in Graphics/Reports	Type	Search
AGE		Age (years)	Age	<input checked="" type="checkbox"/>
WEIGHT	WT	Weight (kg)	Weight	<input checked="" type="checkbox"/>
CRCL		Creatinine Clearance	Continuous	<input checked="" type="checkbox"/>
DSGR		Dose (mg)	DoseGroup (co...	<input checked="" type="checkbox"/>
DSGR		Dose (mg)	DoseGroup (cat...	<input type="checkbox"/>
LGDS		Log(dose)	Continuous	<input checked="" type="checkbox"/>
GRUP		Dose Group	Categorical	<input type="checkbox"/>

PLT Tools

☐ View real-time contents of Editor Output

Figure 22. Tab 2 of the Graphics Editor.

COVARIATES FILE: To access covariates from a Covariates File rather than from NONMEM output tables (FirstRecords or AllRecords), check the "Use Covariates File" box. In

addition, use the **Browse** button to select the name of the Covariates File; then select the delimiter used in that file. Finally, enter the name of the ID column (case-sensitive) in the Covariates file.

Covariates: Covariates for which the user wants graphics are listed here.

Column 1 (Covar. File Term): If covariates for covariate graphics are to be obtained from a Covariates File, enter the exact column header (case-sensitive) in the Covariates File for each covariate to be included in graphics.



Column 2 (NONMEM Term): If a covariate search is to be performed or if covariates from covariate graphics are to be obtained from NONMEM output tables (FirstRecords or AllRecords), enter the column name as listed in the \$INPUT record in the Control Stream.

1. If a Covar. File term matches a NONMEM Term, it is *not* necessary to enter both.
2. If a Covariates File is not selected, column names can be entered into either Column 1 (Covar. File Term) or Column 2 (NONMEM Term).
3. If names entered in Column 1 or Column 2 cannot be matched to actual column headings, a warning message is displayed in the Output window.

Column 3 (Label in Graphics): This column permits the user to select text that is used in the display. For example, an entry of WT in the first column might be associated with "Weight (kg)" in the second column.

Column 4 (Type): This column instructs **PLT Tools** as to the type of covariate. Options are "Continuous", "Categorical", "Age", "Weight", "Height", "Gender", "Race". Default units are: age: years; weight: kg; and height: cm.

Column 5 (Search): This column instructs **PLT Tools** as to whether to include the covariate in a general linear model (similar to a general additive model, GAM).

At startup, a single row is available for entry. Rows can be added or deleted using the  and  buttons at the upper right.

NOTE: ID is automatically added to the covariate list – ID should not be entered by the user as a covariate.

Tab 3: By-Subject Graphics

Many options are available for By-Subject graphics. Most of these are selected in Tab 3 (**Figure 23**).

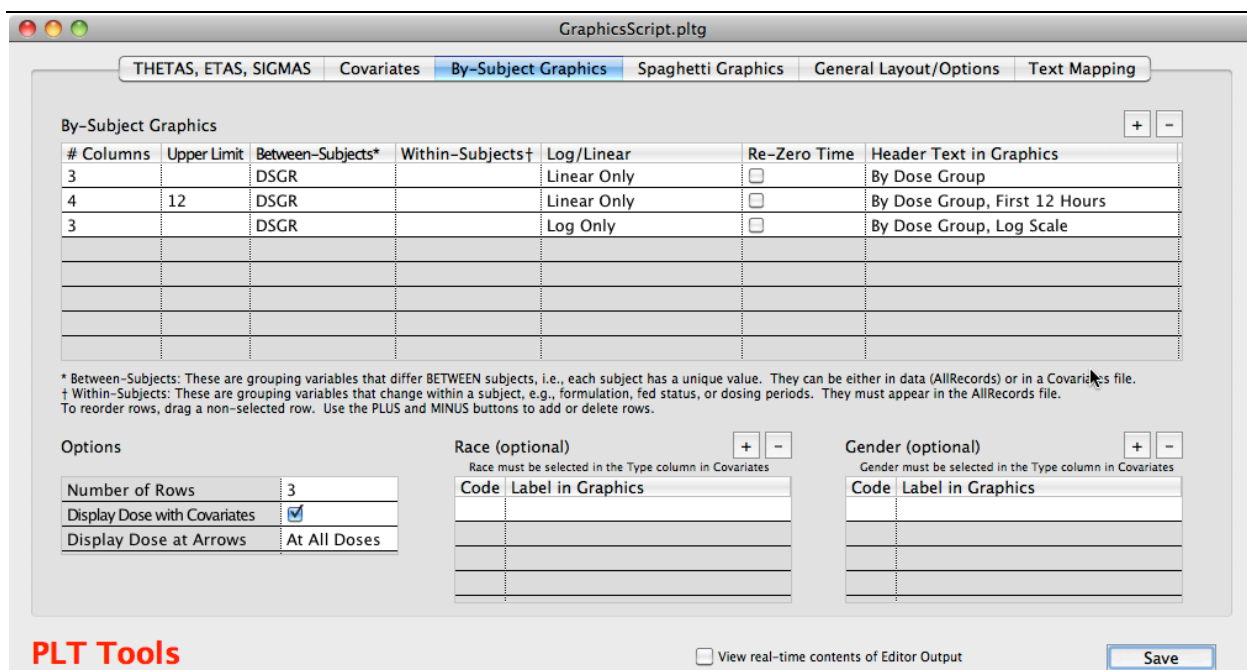




Figure 23. Tab 3 of the Graphics Editor.

Upper area: In the upper area, the user designates sets of graphics. The default is a single set but the user can add or delete sets using the  and  buttons to the upper right.

Column 1: This column indicates the number of columns that appear per page (2, 3, or 4; if no value is selected, the default value 2 is applied).

Column 2: This column indicates whether the upper limit of time should be constrained (e.g., sample may be collected by 48 hours but the analyst wishes to focus one set of graphics on the first 8 hours); a positive integer can be entered here. If no value is entered, each panel auto-scales the x-axis to display all observations > LOQ and at least one terminal BQL value (if present).

Column 3: This column indicates a grouping variable. For example, an analysis might include subjects from two studies. The grouping variable permits the user to display subjects from each study on separate pages with appropriate headers; setup for grouping variables is explained below.

Column 4: This column permits the user to identify a column in the AllRecords file that marks different periods. For example, a clinical trial might involve seven daily doses with intensive sampling following the first and final doses and trough values before each dose. The user could create an input column (which is then output to AllRecords) with the following values: 1 for the first 24 hours; 7 for hours 144-168 (the 24 hours relevant to the final dose); and zero for the remaining values. If the name of this column is entered in the appropriate column in the Graphics Editor, two panels would appear for each subject. The first panel would display the values for the first 24 hours only. The second panel would display the values for hours 144-168; however, the range of the x-axis for that panel would

be 0-24 hours. Headers would appear for each panel indicating the period (see “By-Subjects MultiplePeriods” below).

Suppressing certain periods: Consider a multi-dose study with daily trough samples followed by intensive sampling on Day 7. Enter -1 for all samples before Day 7; then enter a positive value on Day 7 (*e.g.*, 1 or 7). When **PLT Tools** creates graphics for specific periods, it will ignore periods with values < 0; *i.e.*, a graphic will be prepared for the Day 7 values only.

Reset Events: “Reset” events are implemented in NONMEM code by setting EVID to 3 or 4. If the user has done this to permit TIME to be non-monotonic non-decreasing (*i.e.*, to decrease between successive records within a subject), the best approach for preparing graphics in **PLT Tools** is to use “Within-Subjects” grouping, *i.e.*, periods in the dataset. However, if the user has not done so (or has made an error or forgotten to output the period data item to AllRecords), **PLT Tools** recognizes that TIME is not monotonic non-decreasing within a subject and makes special accommodations. These include displaying observations from successive periods with different symbols and predictions with different line types. If the number of successive periods is > 4, these symbols and line types are recycled.

Period	Symbol for Observation (R code)	Line Type for Prediction (R code)
1	circle (1)	solid (1)
2	triangle (24)	dashed (2)
3	inverted triangle (25)	dotted (3)
4	square (22)	long dash (5)

Column 5: This column permits the user to select whether values are displayed on log or linear scales (or both). In addition, if the “take log (either base 10 or natural logs) of both sides” approach has been used, **PLT Tools** can exponentiate the DV, PRED, and IPRED values so that they can be plotted in their original scale.

Entering “other” type records if the “take log of both sides” approach has been used: If the user adds non-dose, non-observation records to the dataset (*e.g.*, to permit smoothing) with EVID = 2 or MDV = 1, the DV values for these entries would typically be zero. Under normal circumstances, the observation for a record with those criteria (DV = 0, EVID = 2 or MDV = 1) is not displayed. However, when these values are exponentiated, the DV value becomes 1 and would be displayed as an “X”. To prevent these DV values from being displayed, enter the value as -9999.

Column 6: This column instructs **PLT Tools** to re-zero times; this is relevant only when the time of the first dose is > zero; if selected, the time of the first dose is subtracted from all time values.

Column 7: This column permits the user to enter text that appears as a header on each page. Examples include “Study GL432”; up to 80 alphanumeric characters are permitted.


Options:

Number of rows: The user selects 2, 3, or 4 rows / page. This applies to all sets of By-Subject graphics.


Label dose as text field: If this box is selected, the magnitude of the first dose is displayed as a text field in each panel; formatting is controlled in the General Layout/Options tab (see below).

Label dose with arrows: Doses (bolus and infusion) in the input data are displayed graphically, either as arrows or vertical lines; exceptions are doses given at a time larger than the upper limit of x (obtained either by auto-scaling or from "Upper Limit" [see above]). Numeric values for doses can be displayed at none of the doses, at the first dose only (appropriate if all doses are identical), or at each dose.

Gender: If the user has provided gender as a covariate (either in the Covariates file or AllRecords) and if "Gender" has been selected as a covariate type in the Covariates area in tab 1, the user can provide names for genders. For example, the Covariates files might contain a column labeled "GNDR". The user enters GNDR as a column header in the Covariates area in tab 1 and selects "Type" as "Gender". If the values entered in the Covariates file are 1 (male) and 2 (female), the user enters 1 and 2 as "Codes" in column 1 of the area in tab 4 and Male and Female in column 2.

NOTE: At startup, only one row is provided; the user will typically need to add a second row using the  button.

Race: If the user has provided race as a covariate (either in the Covariates file, AllRecords, or FirstRecords) and if "Race" has been selected as a covariate type in the Covariates area in tab 1, the user can provide names for races. For example, the Covariates files might contain a column labeled "RACE". The user enters RACE as a column header in the Covariates area in tab 1 and selects "Type" as "Race". If the values entered in the Covariates file are 1 (Caucasian), 2 (African-American), and 3 (Asian), the user enters 1, 2 and 3 as "Codes" in column 1 of the area in tab 4 and Caucasian, African-American, and Asian in column 2.

NOTE: At startup, only one row is provided; the user will typically need to add a second row using the  button.

Displaying A Different Column in a Table: In certain situations, the data analyst wants to display a column other than DV/PRED/IPRED vs. time. For example, when modeling effect vs. time, it may be useful to view concentration in plasma or an effect compartment vs. time to confirm that the pharmacokinetic portion of the model is correct.

In **PLT Tools**, this is accomplished by modifying the "Header Text in Graphics" field in one or more rows in the Graphics Script. In addition, the data must appear in the AllRecords file with the column name Cn (where n is a single digit from 1-9). The header text must be preceded by "PLT PLOT Cn" (omit the quotation marks and replace the lower-case n with a single integer). This supplemental text and a space separating it from the subsequent text will be deleted by **PLT Tools**. For example, the following header text:

PLT PLOT C2 Compartment 2 Concentration

will result in a display of a column named C2 vs. time; the header will be:

Compartment 2 Concentration

NOTE: The y-axis label will be Cn (where n is replaced by the integer provided by the user). The user can supply a replacement y-axis label by supplying a different header text entry. For example, if the header text entry is:

PLT PLOT C2 |NEW Y-AXIS LABEL| Compartment 2 Concentration
the y-axis label would be "NEW Y-AXIS LABEL" (where the user supplies the relevant text). The syntax must be followed exactly: the new y-axis label text must appear between two vertical bars ("|").

Adding Vertical Lines: In certain situations, it might be desirable to display vertical lines in By-Subject graphics to mark events such as a change in a covariate. **PLT Tools** can accomplish this if the `AllRecords` table contains the appropriate data. Examples might include events during a clinical trial such as surgery, detection of anti-drug antibodies, concurrent administration of a second drug, or vomiting.

Create a column in the dataset (and output to `AllRecords`) named either `VLC1`, `VLC2`, `VLC3`, `VLC4`, or `VLC5`. Each subject should have a unique value indicating the `TIME` at which the event occurred (if a subject has no events, enter a value < 0 for that subject). If **PLT Tools** encounters a single positive value for a subject, a vertical line is displayed at that x-value. Line colors are green, red, blue, orange, or magenta for `VLC1`, `VLC2`, `VLC3`, `VLC4`, or `VLC5`, respectively.

If the data item is named `VLC6`, `VLC7`, `VLC8`, `VLC9`, or `VLC10`, the restriction of a unique value is removed. Columns with these names can contain a mixture of 0 and positive values. For any *positive* entry, corresponding entries in the `TIME` column are identified. By-Subject graphics then contain vertical lines at those `TIME` positions; colors are green, red, blue, orange, or magenta, respectively.

Tab 4: Spaghetti Graphics

Spaghetti graphics display the ratio of observed / predicted or the difference between observed and predicted values vs. time. These graphics display time-dependent goodness of fit. Many options are available for spaghetti graphics. Most of these are selected in Tab 4 (**Figure 24**).

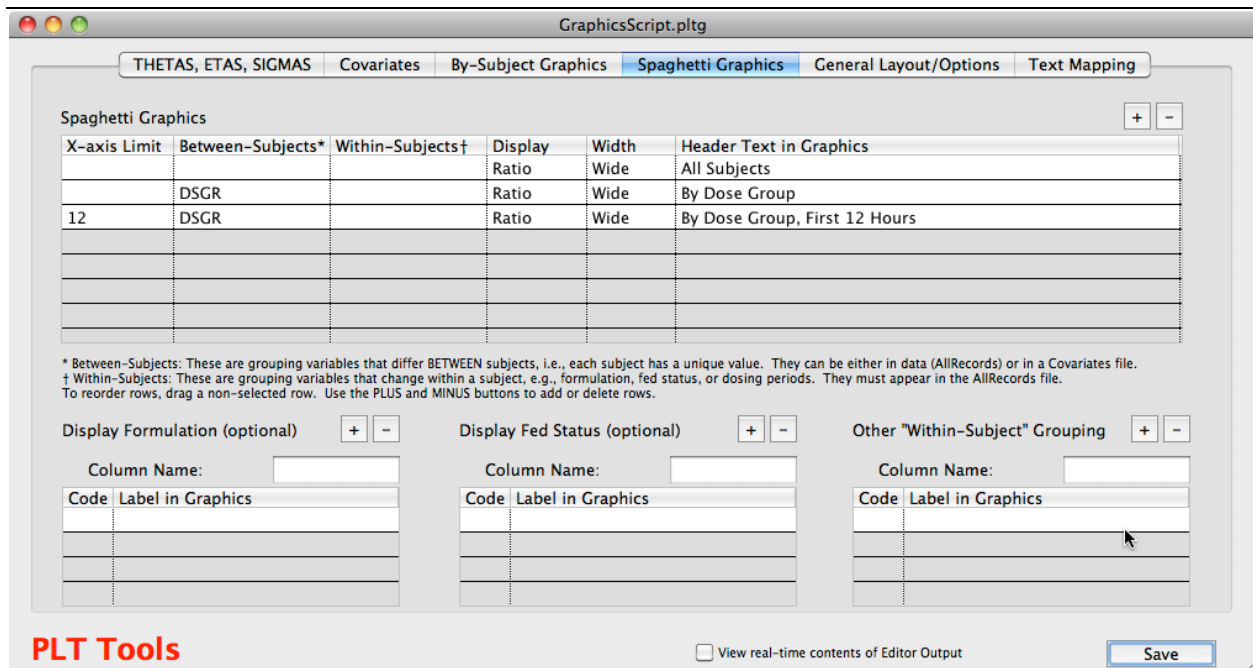




Figure 24. Tab 4 of the Graphics Editor.

Upper area: In the upper area, the user designates sets of graphics. The default is a single set but the user can add or delete sets using the  and  buttons to the upper right.

Column 1: This column indicates whether the upper limit of time should be constrained (e.g., sample may be collected by 48 hours but the analyst wishes to focus one set of graphics on the first 8 hours); a positive integer can be entered here. If no value is entered, each panel auto-scales the x-axis to display all observations.

Column 2: This column indicates a grouping variable. For example, an analysis might include subjects from two studies; the grouping variable permits the user to display subjects from each study on separate pages with appropriate headers; setup for grouping variables is explained below.


Column 3: This column permits the user to identify a column in the AllRecords file that marks different periods. For example, a clinical trial might involve seven daily doses with intensive sampling following the first and final doses and trough values before each dose. The user could create input column (which is then output to AllRecords) with the following values: 1 for the first 24 hours; 7 for hours 144-168 (the 24 hours relevant to the final dose); and zero for the remaining values. If the name of this column is entered in the appropriate column in the Graphics Editor, two pages of graphics would be generated for each. The first page would display the values for the first 24 hours only. The second page would display the values for hours 144-168; however, the range of the x-axis for that page would be 0-24 hours. Headers would appear for each page indicating the period (see "Spaghetti MultiplePeriods" below).

Column 4: This column permits the user to select whether values displayed are ratios or differences (or both, on separate pages).


Column 5: This column permits the user to scale the y-axis for “ratio” plots (“difference” plots are auto-scaled based on data). Choices are “Narrow” (0.2 – 5); “Wide” (0.1 - 10), and “Extreme” (0.01 – 100).

Column 6: This column permits the user to enter text that appears as a header on each page. Examples include “Study GL432”; up to 80 alphanumeric characters are permitted.

Display Formulation: If formulation is a potential covariate in an analysis, the user should enter the column header in `AllRecords` indicating the formulation in two locations, as a Grouping variable in the upper area and as “Column Name” in the “Display Formulation” area. Then, enter the codes for the formulations in the “Code” Column and the corresponding text in the “Value” Column. For example, if the values entered in `AllRecords` are 1 (tablet) and 2 (capsule), the user enters 1 and 2 as “Codes” and Tablet and Capsule as “Values”.

NOTE: At startup, only one row is provided; the user will typically need to add a second row using the  button.

Display Fed Status: If fed status is a potential covariate in an analysis, the user should enter the column header in `AllRecords` indicating the formulation in two locations, as a Grouping variable in the upper area and as “Column Name” in the “Display Fed Status” area. Then, enter the codes for fed status in the “Code” Column and the corresponding text in the “Value” Column. For example, if the values entered in `AllRecords` are 1 (“fed”) and 2 (“fasted”), the user enters 1 and 2 as “Codes” and Fed and Fasted as “Values”.

NOTE: At startup, only one row is provided; the user will typically need to add a second row using the  button.

“Time After Dose”

Spaghetti graphics described previously typically display time on the x-axis; values for time are obtained from the `TIME` data item in `AllRecords`. In certain instances, the user may wish to display time relative to the most recent dose. This is accomplished in the following manner:

1. In the Spaghetti Graphics tab of the Graphics Editor, enter the text “Time after dose” (not case sensitive) as the header text for one or more rows (**Figure 25**). By default, that text will be displayed in the graphic. If the user wishes to suppress that text, enter the following as a portion of the header text: “| |time after dose| |” (the string “time after dose”, all lower case, surrounded on either side by two vertical bars – note the exact spacing and case); if this exact string appears in the Header Text field, it will be deleted from the text that it displayed.

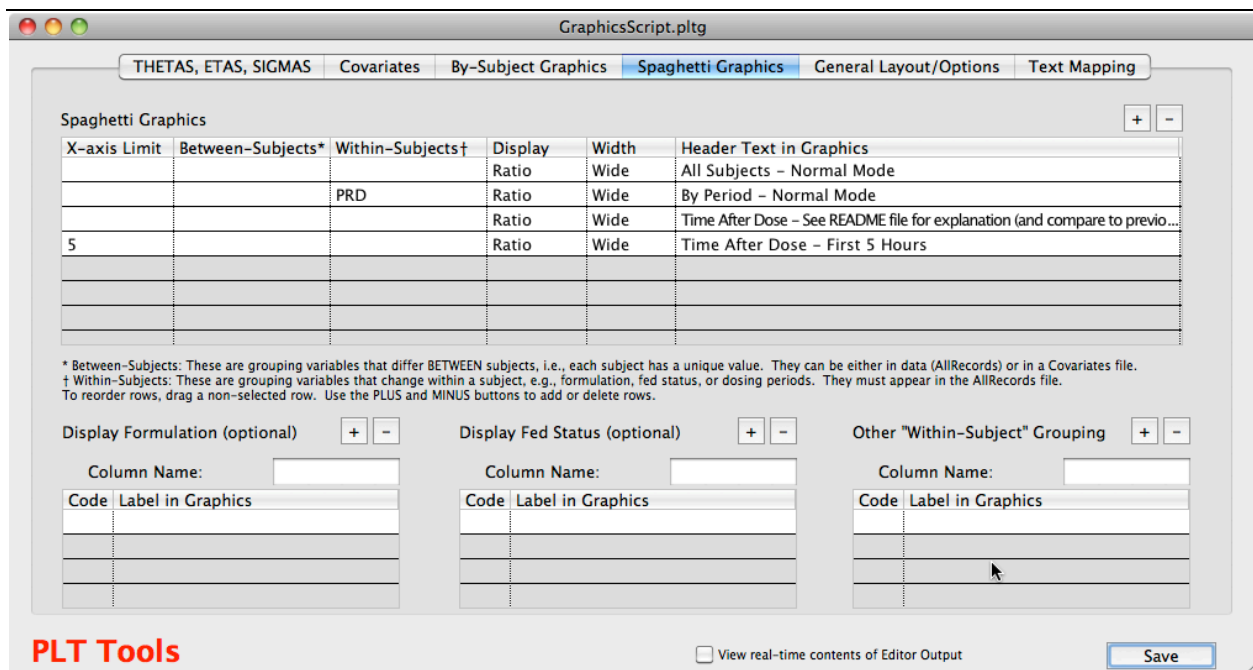


Figure 25. The “Header Text” field in the Spaghetti Graphics tab of the Graphic Editor shows two variants of headers that will create “time after dose” graphics. For the first entry, the text “Time after dose” will appear as the header in the usual manner. For the second header, the text “| |time after dose| |” will be suppressed – the header will be displayed as “Header text”.

2. Time after dose can be calculated using any of three approaches:

a. In the dataset: The user can calculate time after dose during preparation of the dataset. A data item named TAFD must appear in the AllRecords table.

b. By NONMEM: NONMEM can be used to calculate time after dose, using the following code (provided by James Lane, Pharm.D, UCSD):

```
IF(NEWIND.LT.2) THEN
  IFL =0
  TAFD =0.0
ENDIF
IF(EVID.EQ.1) THEN
  TDOS =TIME
  TAFD =0.0
  IFL =1
ENDIF
IF(IFL.EQ.1.AND.EVID.NE.1) TADF =TIME-TDOS
```

A data item named TAFD must appear in the AllRecords table.

c. By **PLT Tools**. If the header text contains “Time after dose”, signaling that a spaghetti graphic should display time after dose rather than time and there is no “TAFD” data item in the AllRecords table, **PLT Tools** will attempt to calculate time after dose. Implicit doses

entered with ADDL/II records will be used in the calculation of dose times. In certain instances, an observation precedes the first dose (*e.g.*, a pre-dose sample obtained before the first dose). In these instances, time after dose will be reported as a negative value relative to that first dose. In other instances, time after dose cannot be calculated. For example, if a dataset includes reset events (EVID equal to 3 or 4), there may be a period during which there are observations but no doses. In this case, time after dose cannot be calculated. A warning will be issued and **PLT Tools** will not display “time after dose” graphics.

“Time after dose” graphics can take advantage of all the other options available for spaghetti graphics in **PLT Tools**, *e.g.*, multiple periods, grouping, time limits, and the display of IDs at each observation time-point. One additional option exists, the ability to display dose number at each observation time-point. Values for dose number can be provided to **PLT Tools** in either of two ways:

1. AllRecords can contain a data item DOSN (typically calculated during creation of the dataset).
2. If “time after dose” is calculated by **PLT Tools** (see above), dose number values are also calculated.*

Note: If all dose numbers are identical, the additional graphic displaying dose number is omitted.

Tab 5: General Layout / Options

General options are selected in Tab 5 (**Figure 26**).

* If the user allows **PLT Tools** to calculate “time after dose” but does not want a display of dose number, add the following code to the NONMEM Control Stream:

a. In the \$THETA block: DOSN = 1

b. Enter DOSN as a data item for AllRecords.txt

Presence of only a single unique value for DOSN will prevent **PLT Tools** from displaying a graphic of dose number.

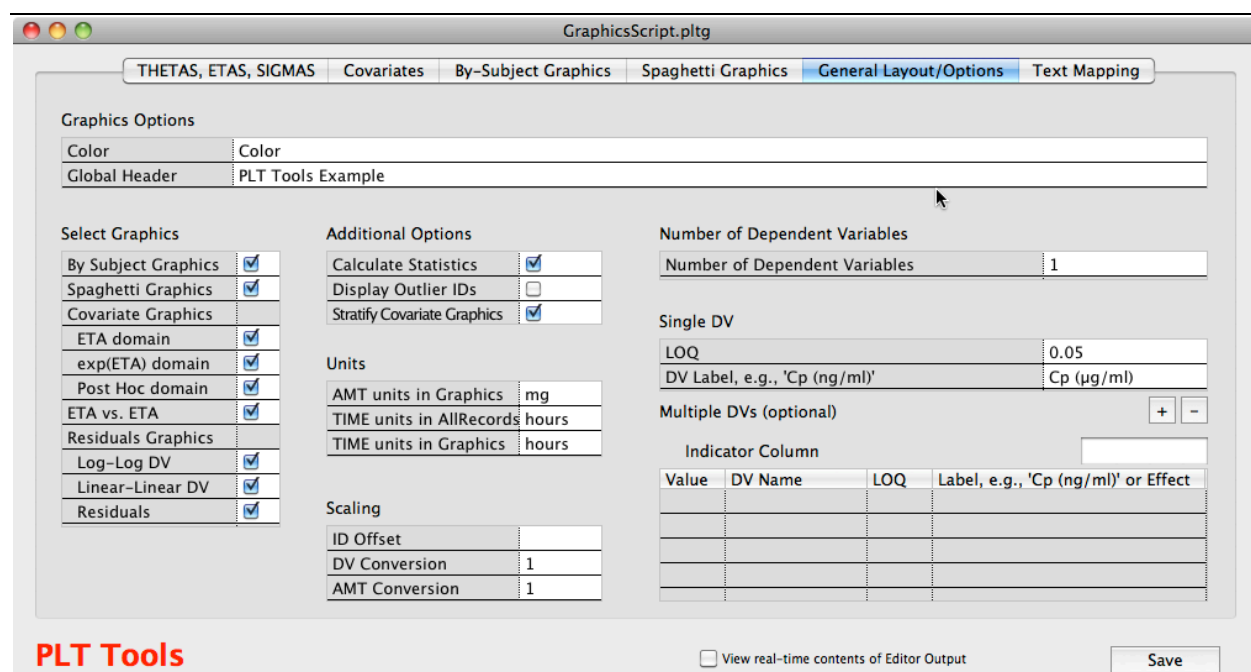


Figure 26. Tab 5 of the Graphics Editor.

Select Graphics: The user can select which sets of graphics are created by checking boxes. By default, all sets are selected.

Additional Options:

Calculate Statistics: If this option is checked, a number of additional statistical calculations are performed and stored in two files:

TEXTFILES/STATISTICS/Statistics.TIMESTAMP.txt and TEXTFILES/GLM-AIC/AIC.TIMESTAMP.txt. The first of these files reports the following:

- median values for each of the parameters
- t* tests comparing *eta* distributions by each of the categorical covariates. For example, if there are three categories for a covariate (A, B, and C), values for *eta* for clearance are compared between groups A and B; A and C; and B and C.
- A GLM (general linear method, similar to the GAM procedure) is performed for each *eta*. Covariates are included in the GLM if they are checked in the GLM column of the first tab of the Graphics Editor.

The second of these files contains the detailed steps of the GLM.

Display IDs: The default versions of goodness of fits (observed / predicted vs. time; observed – predicted vs. time; observed vs. predicted (linear scale or log scale) displays dots at each observation. If the user wants to identify the subject ID associated with one or more points, select this option. A second set of graphics will be created in which dots are replaced with IDs.

Graphics Options:

Color: Most of these graphics use color for emphasis. However, if the user cannot use color, select “Monochrome”. Although graphics continue to display color when displayed with a monitor, additional cues appear in printed versions.

Global Header: The contents of this field appear at the top of every page. However, if the text contained the word “Confidential”, it is deleted. If the user does not create a Graphics Script or if this field is omitted, **PLT Tools** will use the content of the \$PROBLEM statement in the Control Stream (without the \$PROBLEM text) as the Global Header.

Company / Institution: If the user has registered with PLTsoft, the contents of this field appear on the bottom of each page, followed by the text “(Confidential)”.

Units:

AMT units in graphics: In By-Subject graphics, dose amounts can be displayed as a text field. Text entered in this field is used as the units in that display.

Time units in AllRecords: Select the appropriate time unit that applies to the TIME column in the AllRecords file.

Time units in Graphics: Select the appropriate time unit for use in those graphics that display time on the x-axis. Time units in AllRecords are then converted using the appropriate scalar.

Scaling:

ID Offset: The IDs in a dataset (and the corresponding IDs in the Covariate file) may be unwieldy, *e.g.*, the concatenation of a study number, site number and local ID. If so, it may be possible for the user to simplify the ID's by subtracting an offset value. This value is entered as an integer in this field.

DV Conversion: DV values can be scaled by entering a non-zero real number in this field. For example, a value of 0.001 converts between µg/ml and mg/ml.

AMT Conversion: AMT values can be scaled by entering a positive real number (*e.g.*, 0.001) in this field. This permits conversions between units (*e.g.*, dose may be entered in the input in micrograms; scaling by 0.001 converts to mg).

Number of Dependent Variables: NONMEM analyses can involve a single DV (*e.g.*, plasma concentration) or multiple DVs (*e.g.*, a parent and a metabolite or a plasma concentration and an effect). **PLT Tools** can deal with either of these situations by selecting “1” or “>1”.

Single DV:

LOQ: If a single DV is selected and if there is an LOQ for the assay (or for the metric, if DV is an effect), the LOQ should be entered. If there is no known LOQ, the field can be left blank.

DV Units: Text entered in this field will appear in all graphics referring to DV values. A typical entry is “Cp (ng/ml)”.

Multiple DVs: If there are multiple DVs, AllRecords must contain a column identifying the type of DV. This is often accomplished with CMT or FLAG. The name of the indicator column should be entered in the field “Indicator Column”.

Indicator Value: Enter all possible values that appear in the indicator value column that are associated with observations. For example, if a parent compound is modeled with compartment 2 and a metabolite with compartment 4, enter 2 and 4.

DV Names: Enter the names associated with the values that appear in the “Indicator Value” column. These values appear as headers; precise names are recommended.

LOQ: Enter the LOQ for each type of DV. If one or more DVs do not have an LOQ, leave those fields blank.

Units: Text entered in this field will appear in all graphics referring to DV values. Typical entries might be “Cp (ng/ml)”, “Change in BP (mm Hg)”.

Tab 6: Text Mapping

Text mapping options are selected in Tab 6 (**Figure 27**). Text mapping allows the user to customize the text that appears in various panels of the by-subject and spaghetti graphics.

GraphicsScript.pltg

THETAS, ETAS, SIGMAS Covariates By-Subject Graphics Spaghetti Graphics General Layout/Options **Text Mapping**

Between-Subjects Grouping #1

Column Name*	DSGR
Label in Graphics	Dose (mg):

Between-Subjects Grouping #2

Column Name*	
Label in Graphics	

Between-Subjects Grouping #3

Column Name*	
Label in Graphics	

Between-Subjects Grouping #1 + -

Value†	Mapped Text‡

Between-Subjects Grouping #2 + -

Value†	Mapped Text‡

Between-Subjects Grouping #3 + -

Value†	Mapped Text‡

Within-Subject Grouping #1

Column Name ‡	
Label in Graphics	

Within-Subject Grouping #2

Column Name ‡	
Label in Graphics	

Within-Subject Grouping #3

Column Name ‡	
Label in Graphics	

* If these data are in the AllRecords file, enter the exact name that appears in \$TABLE. If PLT Tools obtains the data from a Covariates file, enter the column name from that file.
† Optional: In the left column, enter all possible values in the dataset; in the right column, enter the text to which values are mapped.
‡ Enter the exact name that appears in \$TABLE.

PLT Tools ☐ View real-time contents of Editor Output **Save**

Figure 27. Tab 6 of the Graphics Editor.

Between-Subject Grouping: The user can arrange for subjects to be grouped by grouping variables. For example, an analysis might contain two studies or a single study may contain two groups, young vs. elderly subjects; the “Grouping” options permit separate graphics to be assembled for each of these groups. Up to three different grouping characteristics are supported.

The user informs **PLT Tools** of this grouping in one of two ways. If a covariate file exists and has been identified in the “Covariates File” tab, a column in that file should identify the grouping characteristics (e.g., entries of 1 and 2, corresponding to each of the two studies or age groups). If no covariates file has been identified, the group characteristic must be supplied in the data, then included in the AllRecords file.

Suppressing between-subject labeling: There are instances in which "between-subject" grouping is useful but a label is not. For Label in Graphics, enter "DO.NOT.LABEL" (without the quotation marks).

Grouping:

Column Name: Enter the grouping variable in the field “Column Name”. If a covariate file exists, this is the exact (case-sensitive) column name from the covariate file. If a covariate file does not exist, this is the exact (case-sensitive) column name that appears in the AllRecords file.

Label for graphics: Enter a label, e.g., Study:, in the “Labels for Graphics” field. If this field is omitted but a column name is provided, the default value “Group:” is applied.

Grouping Variable (optional): For each grouping variable, the user can direct **PLT Tools** how to map values in the dataset to text. For example, the dataset might include two groups, young and elderly, coded as 1 and 2. Enter these numeric values in the left column, then the corresponding text (Young, Elderly) in the right column. If the user populates the Column Name but does not populate this area, the numeric values found in the dataset are applied. In certain circumstances, this is desirable. For example, if different groups received doses of 10, 30, and 100 mg, the user could enter “Dose (mg):” as the “Label for Graphics”, then omit the “Grouping Variable” entries. If the dataset included a column containing the values 10, 30, and 100 (and this column is identified in “Column Name”, the resulting graphics would display “Dose (mg): XX” (where XX is the correct value).

Within-Subject Grouping: The user can arrange for separate panels for different portions of a subject's records. For example, a study might consist of a dose at time 0, intensive sampling for 24 hours, followed by another dose at day 7 (followed by 24 hours of sampling); for the purpose of illustration, assume that a sample is also obtained at 48 hours. The default graphic would include a large amount of non-informative white space between hours 24 and 168. Instead, the user can arrange separate panels for each of the 24 hour periods during which sampling is occurring. To accomplish this, the dataset must include a column indicating the within-subject grouping variable. For the scenario described above, several options are available for the values for this column:

1. Assign 1 to all records before 168 hours. Assign 2 to all records starting at 168 hours. The first panel displays values through 48 hours (there are no samples between 48 and 168 hours). The second panel displays values from the second dose (≥ 168 hours).
2. If the goal is to display only the 24 hours for each of these panels, the users can impose a time limit (24 hours) in the appropriate field in the By-Subject or Spaghetti Graphics tabs.

3. Alternatively, the user can assign the value 1 to all records in the first 24 hours and the value 2 to all records related to the second dose. The record at 48 hours is assigned the value -1 (indicating that it should not be displayed in “Multiple Records” displays). When **PLT Tools** selects the data for each panel, each panel will include data for 24 hours and scaling will be automatic.
4. If the user selects the “Multiple Periods” option, it may be desirable to make an additional set of graphics in which that option is not selected (*i.e.*, all data from a subject are displayed in a single panel).
5. A label appears in the graphic identifying the use of this grouping variable. The user can control that label by populating a Within-Subjects grouping variable in the Mapping tab of a Graphics Script. The default text is “Period X” where X is the value in the AllRecords file.

Suppressing within-subject labeling: There are instances in which "within-subject" grouping is useful but a label is not. For example, if there were no samples during an extended period of dosing, then a period of intense sampling, a graphic focused on the period of intensive sampling would be useful. This could be accomplished by creating a "within-subject grouping variable" (*e.g.*, PERIOD), entering a negative integer for records with no samples and a positive integer for the period of intensive sampling. Whether or not a within-subject grouping variable is identified in the Text Mapping tab in a Graphics Script, text will appear in each panel. To suppress this text, populate one of the within-subject group sections. Enter PERIOD (or the user-selected term) for Column Name. For Label in Graphics, enter "DO.NOT.LABEL" (without the quotation marks).

Note: When “Within-Subjects” grouping is selected, **PLT Tools** does not display portions of the data for which there are no explicit doses. If the dataset includes implicit doses (using ADDL and II fields), periods with only implicit doses will not be displayed. If this problem is encountered, add a record to the dataset that contains an explicit dose at the appropriate time (and the value for within-subjects grouping). It will also be necessary to adjust the value for ADDL so as to not replicate the dose.

The fields are populated in the following manner:

Column Name: Enter the multiple-periods variable in the field “Column Name”. This is the exact (case-sensitive) column name that appears in the AllRecords file.

Label for graphics: Enter a label, *e.g.*, Dose #, in the “Labels for Graphics” field. If this field is omitted but a column name is provided, the default value “Period:” is applied.

Displaying Greek Characters in Graphics

Greek characters (*e.g.*, μ) can be displayed in certain fields in graphics. At present, the only fields that are supported as the Global Header (the header that appears at the top of all pages) and DV units (both “Single DV” and “Multiple DV’s” are supported). Instructing **PLT Tools** to display Greek characters requires that the user enter special text (the string GREEK, followed

immediately by a normal letter [either upper or lower case]) in the Graphics Editor. This special text is best explained by examples:

Displaying a Greek character in the Global Header field: The desired user header is “β-Estradiol”. The user enters “GREEKb-Estradiol” in the Global Header field.

Displaying a Greek character in the DV units (DV Label) field: The desired user header is “Cp (μg/ml)”. The user enters “Cp (GREEKmg/ml)” in the Global Header field.

PLT Tools replaces the string “GREEKm” with appropriate code to display the desired Greek character.

Limitations:

1. Some Greek characters are poorly formatted by R. Trial-and-error may be required.
2. Spacing of Greek characters may not be exactly as desired. If so, contact support@PLTsoft.com.
3. At present, Greek characters are supported only in the Global Header and DV units. If the user wishes to display a Greek character in other text fields in Graphics (*e.g.*, Grouping Variables or headers for By-Subject or Spaghetti graphics), this is not supported at present. However, the R code can be modified to accommodate these needs. Contact support@PLTsoft.com.

Changes To Graphics Implemented Outside Of Graphics Editor

The Graphics Editor was designed to accommodate the needs of most pharmacokinetic modeling activities. However, in certain instances, minor, but important, modifications to these graphics have not been accommodated. For example, if a user is analyzing pharmacodynamic data in which time is not an element, By Subject graphics will necessarily contain the word Time in the x-axis label.

Note: If the Control Stream is modified to incorporate these requests, it is critical that the text be entered as a comment, *i.e.*, following a semicolon. If the semicolon is omitted, NONMEM will report an error and will not run.

Note: If the user executes “Graphics Only”, **PLT Tools** reads the stored version of the Control Stream rather than a version in the WorkingFolder. As a result, any changes to Control Streams in the WorkingFolder will not be implemented until “NONMEM + Graphics” is executed. However, if the user wishes to implement changes to graphics as part of the “Graphics Only” procedure, create a file:
GraphicsOptions.txt
in the WorkingFolder. Text in this file should be in the identical format to that described below. The contents of this file are executed during either “NONMEM + Graphics” or “Graphics Only”.

At present, several accommodations are provided.* They are grouped by category. If **PLT Tools** detects a request for any of these changes, it evaluates the text to confirm that the syntax is correct. If it is not correct, an informative error message appears in the Output window. If the user identifies other areas in which such accommodations would be useful, please contact support@PLTsoft.com.

FORMAT / LAYOUT OF GRAPHICS

1. Replace "Time" as the X label: If the Control Stream contains the following text:

```
; PLTTOOLS: TIMEUNITS "XXXX"
```

the text "XXXX" will replace "Time (...)" in the graphics. Note the quote marks around XXXX.

2. Customize labels in By-Subject Graphics: If the user provides certain covariate data (either in the AllRecords file or in a Covariates file) and the user also completes the Covariates area in a Graphics Script, By-Subject graphics display one or more of the following: age, weight, height, race. Age, weight, and height have standard labels ("Age (years)", "Wt (kg)", "Ht (cm)"). The user can customize these labels by entering one or more of the following lines in the NONMEM Control Stream:

```
; PLTTOOLS: CUSTOMAGELABEL xxxxxxxx  
; PLTTOOLS: CUSTOMWEIGHTLABEL xxxxxxxx  
; PLTTOOLS: CUSTOMHEIGHTLABEL xxxxxxxx
```

In addition, if the user provides a data column named "DOSEGROUP" in either the AllRecords, FirstRecords or Covariate files, **PLT Tools** displays "Dose Group: value" in By-Subject graphics. The user can replace the "Dose Group" text in these graphics with customized text by adding the following text to the Control Stream:

```
; PLTTOOLS: DOSEGROUPLABEL xxxxxxxx
```

where xxxxxxxx is a text string (spaces allowed). Should the user wish to display other data in the By-Subject graphics, the user should create a data column named DoseGroup containing the appropriate data, then use this approach to replace "Dose Group:" with the relevant text.

Finally, the user can create one new label field for By-Subject graphics. Enter:

```
; PLTTOOLS: USERCUSTOMLABEL xxxxxxxx
```

in the Control Stream where xxxxxxxx is a text string (spaces allowed, end the string with a colon). The AllRecords or Covariates file must contain a column named CUSTOMLABEL. This can be accomplished by adding one line to the Control Stream:

```
CUSTOMLABEL = covariateName
```

```
; where covariate name is the name of the covariate
```

and adding CUSTOMLABEL to the AllRecords table.

3. Control the upper limit of the y-axis in By-Subject graphics: In by-subject graphics, the upper limit of the y-axis is determined from the observations and predictions (including *post hoc* predictions, if available). The user may wish to impose an upper limit, either to truncate certain values or to ensure that each panel displays the same range of data. To accomplish this, edit

* If users identify additional situations in which these modifications would be useful, please contact support@PLTsoft.com.

the Control Stream so that it contains the following text:

```
; PLTTOOLS: YUPPERLIMIT nnnn
```

where nnnn is the value that the user wishes to impose.

4. Control the lower limit of the y-axis in By-Subject graphics: In by-subject graphics, the lower limit of the y-axis is either zero (linear axis) or calculated from the data and the limit of quantification (log axis). The user may wish to impose a lower limit, either to truncate certain values or to ensure that each panel displays the same range of data. To accomplish this, edit the Control Stream so that it contains the following text:

```
; PLTTOOLS: YLOWERLIMIT nnnn
```

where nnnn is the value that the user wishes to impose.

5. Control the lower limit of the x-axis in By-Subject graphics: In by-subject graphics, the lower limit of the x-axis is typically zero. The user may wish to impose a lower limit, either to truncate certain values or to ensure that each panel displays the same range of data. To accomplish this, edit the Control Stream so that it contains the following text:

```
; PLTTOOLS: XLOWERLIMIT nnnn
```

where nnnn is the value that the user wishes to impose.

6. Scale the y-axis of By-Subject graphics to the data: In by-subject graphics, the lower and upper limits of the y-axis are determined as described above. The user may want linear graphics in which the lower limit is markedly different from zero, *e.g.*, if pharmacodynamic data with a narrow data range are being plotted. To accomplish this, edit the Control Stream so that it contains the following text:

```
; PLTTOOLS: YLIMITSFROMDATA
```

If either of the previous two options (YUPPERLIMIT, YLOWERLIMIT) is selected, this option is ignored.

7. Scale the y-axis of linear Spaghetti graphics to the data: In linear spaghetti graphics (observed – predicted), the lower and upper limits of the y-axis are determined so as to minimize space while maximizing the fraction of data displayed. In some instances, this can truncate data of interest. To ensure that all data are displayed, edit the Control Stream so that it contains the following text:

```
; PLTTOOLS: LINEAR.LIMITS.FROM.DATA
```

8. Fix the lower and upper limits of spaghetti plots in the linear domain: If the user displays “Difference” (rather than “Ratio”) for spaghetti plots, the lower and upper limits for the y-axis are determined using an algorithm that attempts to minimize both whitespace and the fraction of data that are out of bounds. To override this algorithm, edit the Control Stream so that it contains the following text:

```
; PLTTOOLS: SPAGHETTIDIFFBOUND nnnn
```

where nnnn is a real number. The y-axis will then extend from –nnnn to +nnnn.

9. Establish a minimum span for the x-axis for By-Subject graphics: The x-axis span for each By-Subject graphic is determined so as to minimize white-space. A fixed value can be established using “X-Axis Limit” option in the Graphics Editor. If the user wants to establish a minimum span (but still allow a larger value if indicated by the data), this is accomplished by

adding the following comment to the Control Stream:

```
; PLTTOOLS: FORCESPAN N.N
```

where N.N is a real number or an integer.

10. Alter lower axis limits for log-log observed vs. predicted graphics: If the user selects a Graphics Script and populates the LOQ field with a non-zero value, log-log observed vs. predicted graphics are scaled to include the LOQ, even if no observed or predicted values are in that range. Adding:

```
; PLTTOOLS: LOG.LIMITS.FROM.DATA
```

to the Control Stream alters the procedure so that the graphics are scaled entirely to the data.

11. Suppress certain doses in graphics: In certain instances, doses may entered in the data for two purposes, e.g., a dose of drug and a “dose” to load a pharmacodynamic compartment. Under normal circumstances, **PLT Tools** will display both sets of doses in BySubject graphics. However, if the intent is to display only the drug (and not the loading dose), this can be accomplished with the following text in the Control Stream:

```
; PLTTools SUPPRESS.CMT.DOSE nn
```

where nn is an integer indicating the compartment for which the dose should be suppressed from graphics.

12. Visual Predictive Check: Fix the Upper Limit of the X or Y-axis: **PLT Tools** attempts to identify appropriate limits for both axes. However, there may be instances in which these axis limits are not optimal. For example, if a “time after dose” graphic contains only one sample > 1.0 day post-dose and that sample is at 10.0 days post-dose, the x-axis limit would be ~ 10 days and the bulk of data would be compressed in the left portion of the graphic. To impose a limit for the VPC x-axis, add the following code:

```
; PLTTOOLS: XAXIS.LIMIT AAA
```

where AAA is a positive real number; if a time-after-dose VPC is created, the same limit will be imposed. To impose a different limit a time-after-dose graphic, add the following code:

```
; PLTTOOLS: XAXIS.LIMIT AAA BBB
```

where BBB is a positive real number. To impose limits on the y-axis, add the following code:

```
; PLTTOOLS: YAXIS.LIMIT AAA
```

```
; PLTTOOLS: YAXIS.LIMIT AAA BBB
```

Elements in these strings can be separated by spaces and/or tabs.

At present, it is impossible to impose a limit only on the time-after-dose graphic.

STATISTICS

1. “Always run GLM”: During multiple iteration runs (likelihood profile, bootstrap, jackknife), **PLT Tools** typically does not perform statistical analyses (e.g., calculation of median parameter values; *t* tests comparing *eta* values between covariate groups; GLM) after the initial estimation run. If the user wishes to perform these statistical tests, edit the Control Stream so that it contains the following text:

```
; PLTTOOLS: ALWAYSRUNGLM
```

2. Add statistics for covariate graphics in the parameter domain: Covariate graphics in the *eta* domain (i.e., *post hoc etas* vs. covariates) display statistics (e.g., *P* and *r* values for a linear

regression for continuous covariates). By default, these statistics are not displayed in the parameter domain (*i.e.*, parameters vs. covariates). To display these statistics in graphics in the parameter domain, add the following comment to the Control Stream:

```
; PLTTOOLS:      ADDSTATS
```

SIMULATION

Outside the Context of a VPC:

1. Do not create graphics for simulations when *NPROBLEM* is > 1: When simulation is performed with *NPROBLEM* > 1, **PLT Tools** creates graphics for each sub-problem. This can be time-consuming and the user may not want these additional graphics. To suppress creation of these graphics, add the following comment to the Control Stream:

```
; PLTTOOLS:      NO.GRAPHICS.FOR.SIMULATION
```

2. Do not create sub-tables for simulations when *NPROBLEM* is > 1: When simulation is performed with *NPROBLEM* > 1, **PLT Tools** creates tables for each sub-problem (in addition to the “master table” that contains all data). This can be time-consuming and the user may not want these additional graphics. To suppress creation of these graphics, add the following comment to the Control Stream:

```
; PLTTOOLS:      NO.TABLES.FOR.SIMULATION
```

Within the Context of a VPC:

1. Create graphics for simulations: **PLT Tools** does not create graphics for each sub-problem during a VPC. To allow creation of these graphics, add the following comment to the Control Stream:

```
; PLTTOOLS:      MAKE.GRAPHICS.FOR.SIMULATION
```

2. Create sub-tables for simulations: **PLT Tools** does not create tables for each sub-problem (in addition to the “master table” that contains all data. To allow creation of these graphics, add the following comment to the Control Stream:

```
; PLTTOOLS:      MAKE.TABLES.FOR.SIMULATION
```

TIME AFTER DOSE

1. Calculate “Time After Dose” in AllRecords: If the user wants a visual predictive check to display “time after dose” on the x-axis, this can be accomplished in one of two ways. First, the user can create a data item named TAFD. If this data item is output to the AllRecords table, two sets of visual predictive check graphics will be created, one with TIME on the x-axis, the other with time after dose on the x-axis. If the user has not created this data item, **PLT Tools** will attempt to do so (and create the appropriate graphics) if the Control Stream contains the following code:

```
; PLTTOOLS:      CREATE.TIME.AFTER.DOSE
```

VISUAL PREDICTIVE CHECK

1. Suppress VPC graphics in the log domain: Graphics for a visual predictive check are normally created in both the linear and log domains. To suppress graphics in the log domain, add the

following comment to the Control Stream:

```
; PLTTOOLS:      VPCLINEARONLY
```

2. Customized percentile bands for Visual Predictive Check: The graphic for a Visual Predictive Check displays the following percentile bands by default: 2.5, 5, 25, 50, 75, 95, 97.5. If the user prefers alternative percentile bands, these can be selected by adding the following comment to the Control Stream:

```
; PLTTOOLS: VPCPERCENTILES nn1 nn2 nn3
```

where nn1, nn2, and nn3 are real numbers. Avoid unnecessary characters such as commas. Up to seven values can be entered (their order is not important; they will be sorted).

3. Controlling the number of bins: By default, the number of bins for calculation of percentiles for a VPC, a prediction-corrected VPC, and a standardized VPC is the smaller of the number of unique values of the independent variable (typically, time) and 50.

Three options are available to override this:

VPC: Add the following comment to the Control Stream:

```
; PLTTOOLS:      NBins nnn
```

Prediction-corrected VPC: Add the following comment to the Control Stream:

```
; PLTTOOLS:      pcVPCBins nnn
```

Standardized VPC: Add the following comment to the Control Stream:

```
; PLTTOOLS:      StandardBins nnn
```

In each case, nnn is a positive integer; nnn should be smaller than both the number unique values of the independent variable and 50). NBins, pcVPCBin, and StandardBins can be separated from the previous and following fields by tabs and/or spaces.

4. Controlling y-axis limits: By default, the lower bound for a pcVPC is zero. The user can override this by adding the following comment to the Control Stream:

```
; PLTTOOLS:      LowerBound nn.nn
```

where nn.nn is a real number. LowerBound can be separated from the previous and following fields by tabs and/or spaces.

By default, the upper bound for a VPC and pcVPC is determined from the data (using a complicated algorithm). The user can override this by adding the following comment(s) to the Control Stream:

```
; PLTTOOLS:      VPC.UpperBound nn.nn
```

```
; PLTTOOLS:      VPCpc.UpperBound nn.nn
```

where nn.nn is a real number. VPC.UpperBound and VPCpc.UpperBound can be separated from the previous and following fields by tabs and/or spaces.

5. Suppress VPC Observation Lines: **PLT Tools** attempts to calculate percentiles 5, 50, and 95 of the observations and display these as dashed lines. These lines can be suppressed by adding the following text to the Control Stream:

```
; PLTTOOLS:      SUPPRESS.VPC.OBSLINES
```

6. *Suppress VPC Text*: **PLT Tools** attempts to add text to the graphic to quantify dosing and sampling regimens. This text can be suppressed by adding the following to the Control Stream:

```
; PLTTOOLS:      SUPPRESS.VPC.TEXT
```

7. *Keep BQL Values*: **PLT Tools** normally removes BQL values that were included in the analysis (e.g., $DV > 0$ and $EVID$ or $MDV = 0$) from VPC calculations. These values can be included in the calculations by adding the following to the Control Stream:

```
; PLTTOOLS:      KEEP.BQL
```

8. *Performing pcVPC in the log domain*: By default, pcVPC is performed in the linear domain. Adding the comment below to the Control Stream indicates that the pcVPC should be performed in the log domain:

```
; PLTTOOLS:      log.pcVPC
```

This option is presently not implemented. If it is required, contact support@PLTsoft.com.

Tornado Plots

Tornado plots display the range of parameter under a variety of conditions, e.g., *post hoc* estimates or the impact of a particular covariate on that parameter. An example is displayed in **Figure 28**. **PLT Tools** can create tornado plots. However, the user needs to provide information regarding the content and appearance of the graphic. This information is provided by the user in the form of a text file named `TornadoCommands.txt` that is located in the folder `POSTPROCESSING/TORNADO.SCRIPT`.

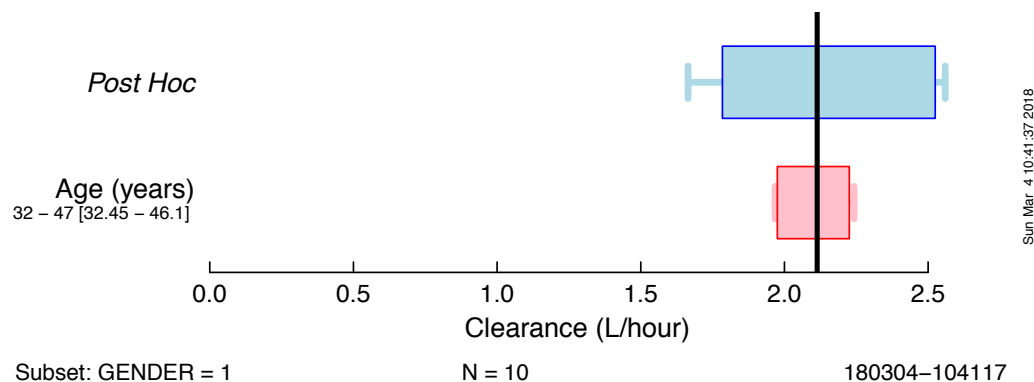


Figure 28. A sample tornado plot.

Creating the tornado graphic requires that the user do all of the following:

1. A Graphics Script must exist and it must contain the parameters and covariates relevant to this graphic
2. A `FirstRecords` file must exist for that run and it must contain several elements that might not normally exist. For example, consider a `$PK` block containing the following code:

```
AGEFACTOR = 1 + (AGE - MEDIANAGE) * THETA(2)
TVCL      = THETA(1)
CL        = TVCL * AGEFACTOR * EXP(ETA(1))
```

The `FirstRecords` file must contain each of the following:

```
TVCL      # population value
```

CL # post hoc values
AGEFACTOR # effect of age on CL

Note that both TV (typical value) of clearance (*i.e.*, before incorporation of ETA) and clearance (after incorporation of ETA) must appear in the FirstRecords file.

The contents of TornadoCommands.txt are:

TAG	Required	Refers to File	Column or Context
TV	yes	FirstRecords	population value, in this case TVCL
POSTHOC	yes	FirstRecords	<i>post hoc</i> values, in this case CL
COVAR	yes	Covariate/ FirstRecords	First entry: covariate values Second entry: effect of covariate Third entry: Label for covariate
SUBSET	no	Covariate	First entry: covariate to select a subset Second entry: covariate value to apply
LABEL	yes	—	Label to appear in graphic (text)
PLUS	no	—	If the covariates are entered into the model by addition rather than multiplication, <i>e.g.</i> , $CL = (TVCL + AGEFACTOR + WTFACOR) * EXP(ETA(1))$
KEEP.NEG	no	Covariate	Do not replace covariates values < 0 with median of the remaining values
PERCENTILE	no	—	Calculate a confidence interval other than 5-95

A sample file contains the following:

```
PERCENTILE  2.5           ; calculate CI from 2.5-97.5 (rather than 5-95)
SUBSET                               ; subset includes all subjects
SUBSET      PLTGENDER      1       ; contains subjects with gender=1
SUBSET      PLTGENDER      2       ; contains subjects with gender=2
TV          TVCL           ; column containing typical value
POSTHOC     CL             ; column containing post hoc values
LABEL Clearance (L/hour)
COVAR PLTAGE      CLAGE Age (years) ; see footnote
```

Footnote: COVAR tags are followed by three entries:

1. Column name in the Covariates file that contains the covariate, *e.g.*, the subjects' weight
2. Column name in the FirstRecords file that contains the effect of that covariate on the typical value, *e.g.*, WTFCTR (see sample NONMEM code above).
3. The label that appears in the graphic

Note that **PLT Tools** converts the names of "reserved" covariates in the covariates file (but only if a covariates files is created by **PLT Tools**). For example, if the FirstRecords file contains SEX and the Graphics Script identifies this as Gender (in the Covariates tab), it is necessary to use PLTGENDER rather than SEX in the TornadoCommands.txt file. Other

reserved covariate names are age (PLTAGE), weight (PLTWT), height (PLTHT), and race (PLTRACE).

Covariate values less than 0: Some data analysts code missing covariate values with a negative number, *e.g.*, -1 or -99. **PLT Tools** recognizes these as missing values and does not include them in certain analyses. The default for the tornado plot is to exclude these values, replacing them with the median of the remaining values. If the negative values should be preserved, add "KEEP.NEG" (without quotes) in the TornadoCommands.txt file.

The following rules **must** be followed in the creation of the TornadoCommands.txt file:

1. Field separators: Each of the entries must be separated by a single tab. No spaces or tabs should appear to the left of the tag.
2. If a semicolon (;) appears on any row, the semicolon and all text that follows is deleted. Therefore, semicolons cannot should be used only in comments.

Labels are on y-axis are:

1. *Post hoc* appears first
2. For each covariate, the names of the covariate appears above numeric values. The numeric values are range, followed by percentile values for that covariate (*e.g.*, percentile 5 — percentile 95, unless the user has selected other percentiles using the PERCENTILE command in the TornadoCommands.txt file).

Tornado graphics are created under two different circumstances:

1. Once a TornadoCommands.txt file exists, every time NONMEM is run, **PLT Tools** attempts to create the graphic.
2. The user can request that a tornado plot be created by clicking on the Tornado Plot button in the "Prepare a Report" tab.

Additional Graphics

Certain graphics are not controlled *via* the Graphics Editor. Instructions for creation of these graphics are provided here.

Data Checkout

When an analysis is applied to a dataset for the first time, it is useful to examine the data to ensure that observations have been entered into the dataset at reasonable times in relationship to doses and that the magnitude of observations is consistent between subjects (*e.g.*, the same units). This can be facilitated *via* several graphics available with **PLT Tools**. Each of these graphics is created the first time that graphics are created within a Project Folder or when a Graphics Script is populated. If dataset changes, when NONMEM is run again, these files are re-generated.

Files are is located in the INITIAL.DATA.CHECK folder within PROJECTFOLDER.

Observations By Subject

This graphic displays a single panel for each subject. The x-axis is time; the y-axis is DV. Each observation with MDV = 0 is displayed with a circle; values are connected by a line. Each time at which a value for AMT > 0 appears in the data is marked with an arrow.

Composite Observations

This graphic displays composite data on each of linear and log scale. Each panel displays the data for all subjects with circles for each observation (MDV = 0) connected by thin lines. If the table AllRecords.txt contains a column "DSGR", it is assumed that this column indicates the dose group (with the value indicating the dose). If so, the following happens:

1. The composite graphics described above display different colors for each dose group.
2. A second set of graphics (both linear and log scales) displays values for each dose group.

Dose-Normalized Observations

This graphic is created only if AllRecords.txt contains a column "DSGR". It is assumed that this column indicates the dose group (with the value indicating the dose). Composite graphics are created (one with a linear scale; one with a log scale) in which dose-normalized concentrations ($DV / DSGR$) are displayed; values for each subject are connected by a line. Each dose group is represented with a different color.

Covariate Summary

Various graphics are prepared to display continuous and categorical covariates; in addition, covariates are summarized in table format (saved with a comma-separated values [CSV] format). These graphics and tables are prepared only when a Graphics Script has been populated.

Covariate vs. Covariate

Pairs of covariates are displayed against each other. A linear regression (with the corresponding r and P values) and a smoother are displayed. This graphic is created only once, the first time that graphics are created within a Project Folder. The file is named CovariateVsCovariate.pdf and is located in the Graphics folder within Project Folder. If the covariate data set changes, delete the existing file; the next time that graphics are created by the Controller, this graphic will be generated again. This graphic is prepared only when a Graphics Script has been populated.

Visual Predictive Check

If the user has selected Visual Predictive Check in the Project Controller window, **PLT Tools** will attempt to create a graphic displaying the Visual Predictive Check. The user is responsible for constructing the Control Stream and the dataset in an appropriate manner. Recommended steps include:

Change to the Control Stream:

1. Enter the final estimates from a successful estimation run as the initial values for each *theta*, *eta*, and *sigma*.
2. Delete (or comment out) the \$EST record
3. Create a \$SIM record and select the ONLYSIMULATION option (from NONMEM Help: "PRED-defined data items in tables and scatterplots will be computed using simulated etas and initial thetas.").
4. If the number of subjects in the dataset is small, use the SUBPROBLEMS option to obtain additional simulated values at each timepoint. For example, if the dataset contains 50 subjects and the user wants 1000 predictions at each timepoint, enter SUBPROBLEMS=20. A sample \$SIM record is:

```
$SIM ONLYSIM SUBPROB=20 (1111111)
```
5. Typically, the \$INPUT and \$DATA records should **NOT** be changed.

Changes to the Dataset: The dataset should include the original DV column. These values will be replaced with simulated values by NONMEM. **PLT Tools** will access NMTRAN's FDATA file to obtain the DV values for graphics.

PLT Tools performs the simulations, collects the observations at each timepoint, calculates percentiles at each timepoint, then prepares graphics (see below). The user is responsible for correct implementation of the NONMEM code. **PLT Tools** performs limited error checking of the dosing and sampling regimens: text appears in the graphic indicating if the dosing and/or sampling regimens are identical for all subjects.

Standardized Visual Predictive Check

A standardized visual predictive check (Wang DD, Zhang S. Standardized Visual Predictive Check Versus Visual Predictive Check for Model Evaluation. J Clin Pharmacol 2012; 52:39-54) is created. If a group variable (VPCG) is provided (see above), a second graphic stratifies groups by color.

Prediction-Corrected Visual Predictive Check

A prediction-corrected visual predictive check (Bergstrand *et al.* Prediction-Corrected Visual Predictive Checks for Diagnosing Nonlinear Mixed-Effects Models. AAPS 13:143, 2011) is created.

Between-Subject Grouping Within a VPC

In certain situations, a visual predictive check may require modification. For example, the user may have two or more markedly different dose groups, pooling of which may limit information about the model's ability to predict concentrations relevant to a particular dose. **PLT Tools** offers the opportunity for the user to create VPC graphics for subgroups. This is accomplished in the following manner:

1. The dataset must contain a column indicating between-subject grouping. For example, in a dose-escalation study in which subjects received doses of 10, 30, or 100 mg, a column might indicate the dose for that session (or a code identifying the group).
2. The \$INPUT record must identify that column with the name VPCG (*i.e.*, VPC group). This column probably already exists in the dataset with a different name, *e.g.*, DOSEGR (dose group). If the DOSEGR term is used in the \$PK block, the simplest approach is the following:
 - a. in \$INPUT, change DOSEGR to VPCG
 - b. in \$PK, add the record: DOSEGR = VPCG
3. The \$TABLE record for AllRecords must include VPCG.

If **PLT Tools** detects a column in each of FDATA and AllRecords named VPCG, a second set of graphics will be created. These graphics will display each group on a separate page; a header will identify the group.

Within-Subject Grouping Within a VPC

PLT Tools can stratify VPC graphics by a within-subjects marker, *e.g.*, a dosing period. This is accomplished in the following manner:

1. The dataset must contain a column indicating within-subject grouping. For example, in a food effect study, one dosing session could have the value 1 for all fasted records and 2 for all fed records; in a dose escalation study, a column might indicate the dose for that session.
2. The \$INPUT record must identify that column with the name VPRD (*i.e.*, VPC period). This column probably already exists in the dataset with a different name, *e.g.*, FORM (formulation). If the FORM term is used in the \$PK block, the simplest approach is the following:
 - a. in \$INPUT, change FORM to VPRD
 - b. in \$PK, add the record: FORM = VPRD
3. The \$TABLE record for AllRecords must include VPRD.

Nominal Time vs. Actual Time

In certain situations, use of actual times (rather than nominal times) will influence the results of a VPC. Consider a study in which a sample is to be obtained 4 hours post-dose from 1000 subjects. In all but one subject, the sample is obtained at that time; the remaining subject is sampled at 4.1 hours. If the simulation is performed using actual times (and SUBPROBLEMS = 1), only a single prediction will be obtained at 4.1 hours, the remainder at 4 hours. The percentile graphics calculated for the VPC may be misleading as a result of this “unbalanced” design.

This problem can be overcome by making all predictions at nominal times. This requires that the dataset contain nominal times. However, associating the observations with the nominal times is misleading (*e.g.*, the sample obtained at 4.1 hours would be recorded as having been obtained at 4 hours).

PLT Tools addresses this problem using the following approach:

1. The dataset must contain both nominal and actual sample times (same units).
2. Nominal sample times are identified in \$INPUT as TIME. As a result, predictions are obtained at nominal times.
3. Actual sample times are identified in \$INPUT as XTME.
4. The \$TABLE record for AllRecords must list both XTME and TIME (column order is not important). An alias is permitted for TIME.
5. When NONMEM creates tables, DV values (observations) are replaced with simulated values.

PLT Tools obtains the original DV values (observations) from FDATA.

6. If **PLT Tools** detects the presence of XTME in FDATA, XTME (actual time) will be used for display of observations; TIME will be used for display of predictions. A warning will be displayed in the OUTPUT window and a footnote will appear in the graphic.

Likelihood Profile

If the user has selected Likelihood Profile in the Project Controller window, **PLT Tools** will attempt to create a graphic appropriate for that analysis.

Creating Your Own Graphics

The suite of graphics produced by **PLT Tools** was developed over many years to address the needs for most NONMEM projects. However, on occasion, users may need to create specialized graphics. For users familiar with R, this is accomplished using the following approach:

1. In PROJECTFOLDER, make sure that there is a folder named USERSCRIPTS
2. In that folder, create a file named UserDefinedScript.R.
3. That file should contain instructions for R. These commands must have the correct syntax.
4. The following “objects” are available to the user-defined script:
 - A. ALL – the contents of the AllRecords file
 - B. ETAS – the contents of the FirstRecords file (if one exists) or the first records for each subject from the AllRecords file
 - C. DEMO – the contents of the Covariates file (if one exists) or the first records for each subject from the AllRecords file.
 - D. FILEID – the 13-character time-stamp
 - E. GlobalHeader – the header that appears on each page of graphics

Sample files are available in the **PLT Tools** folder in the subfolder USERSCRIPTS.

The user will need to work iteratively with **PLT Tools** to develop his/her own script. For example, initial content of the user-defined script may merely list all available objects (“ls()”). Next, the user can write each of these objects to a file or print them to the Output window using the commands “cat” or “print”. Once the user is familiar with the objects, he/she can write code appropriate for the creation of graphics or statistical analysis.

If the user wishes to access other objects (e.g., the list of parameters generated from the Graphics Script), PLTsoft may be able to alter the internal code of **PLT Tools** to provide access to these objects.

If the user is not familiar with R (i.e., and is therefore unable to create the script), scripts can be prepared by PLTsoft on a consulting basis. Contact support@PLTsoft.com.

Note: If there is a user-defined script in the USERSCRIPTS folder, **PLT Tools** will execute that code from any WORKINGFOLDER and any Control Stream within the PROJECTFOLDER. If the user wishes to prevent **PLT Tools** from executing a user-defined script, add the following line to a Control Stream:

`; NOUSERGRAPHICS`

Using the Summarize / Report Tool

The Summarize / Report tool (**Figure 29**) is designed to facilitate summarization of individual NONMEM runs, comparison of one or more pairs of NONMEM runs, and preparation of a report. The Summarize / Report tool consists of two elements, Summarize / Compare and Assemble Report. Assemble Report performs the same tasks as Summarize / Compare; in addition, it creates a rich-text (RTF) document that can be opened in either Microsoft Word or Open Office.

Prepare a Report

Prepare a Report can perform its tasks on either a Run Listing or from a single run. To prepare a Run Listing, see "Using the Run Listing Editor"; using the pull-down menus and checkboxes, the user can then select to use all of the Run Listing or a subset (the BASE, FINAL, or OTHER runs or any combination thereof). Alternatively, the user can select a single run # (timestamp) by selecting "Single Run" from the Source pull-down, then selecting the timestamp from the pull-down labeled 'If "Single Run". Then, the tool performs several tasks for each selected run:

1. A brief summary (identical to the Brief Summary prepared at the completion of each run, except that this version is stored as .txt rather than .pdf).
2. A text comparison between the "New Model" and the "Old Model". Details are provided below. This task is not performed if only a single run is selected.
3. Certain details of comparisons are assembled into a table and a flowchart. Details are provided below.

Flowcharts: If the Run Listing contains more than one entry, the Report Tool in **PLT Tools** assembles a flowchart showing the conduct of the analysis. This flowchart is created in both PDF and JPEG formats and can be embedded into Word documents. In certain instances, Word rasterizes embedded PDF graphics, blurring the normally crisp image. Should this occur, the image can be optimized by opening the PDF document, converting to JPEG format, then inserting the JPEG version. This JPEG version typically appears better than the JPEG version created in R.

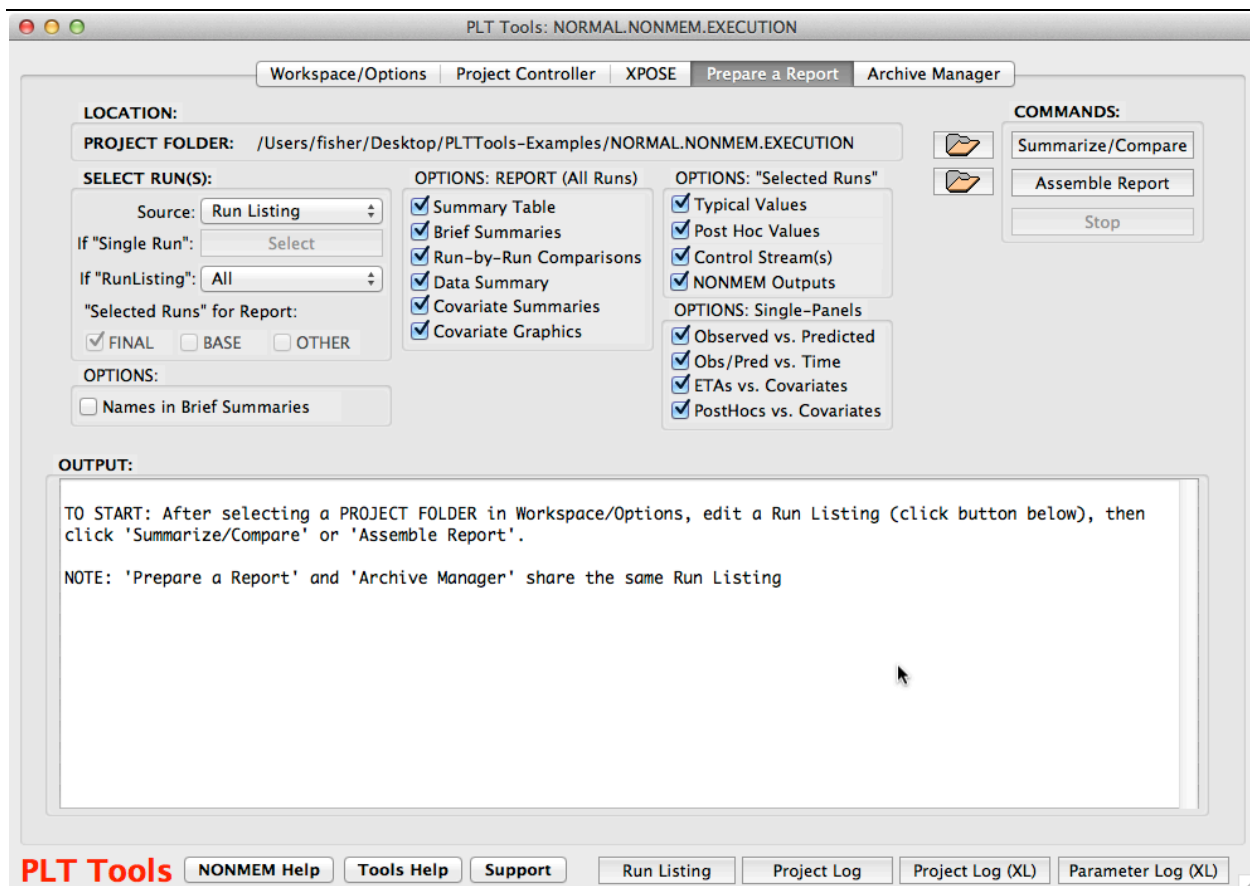


Figure 29. The Prepare a Report tool.

One option is available for the Prepare Report tool: “Add THETA and OMEGA names to Brief Summaries”. If this option is selected, **PLT Tools** examines the archived Graphics Script for that run to determine if THETA# values were entered *via* the Graphics Editor. If so, these names, when available, appear in the left column of the output. In addition, if THETA# values were entered, the names corresponding to OMEGA values are displayed in the left column of the output. If this option is not selected, the THETA and OMEGA sections of the Brief Summary documents do not contain parameter names.

To facilitate assembly of the Run Listing, the user can click **View Project Log**. The first column in the Project Log is the time stamp associated with every run conducted in the Project Folder. The third column is the name of the Control Stream that the user selected for that run. This is useful if the user names each Control Stream in an informative manner. The fourth column is the contents of the \$PROBLEM record from the Control Stream. This is useful if the user updates the \$PROBLEM statement in an informative manner. If the user always works from the same Control Stream and does not update the \$PROBLEM statement, the Project Log is not useful.

The Summarize / Compare procedure is conducted in the following manner:

1. If the pair includes two runs, the objective function of each run is determined, the difference in the objective function is calculated, and if the “change in degrees of freedom” is reported, a *P* value for the likelihood ratio test is calculated. If comments appear in the individual Control Streams (a “comment” is a line preceded by “;C”), these are reported. Then, a line-by-line comparison of the two Control Streams is performed: all differences (other than white-space) are reported. If only one member of a pair is identified (see row 1 in the example), only values relevant to that run are reported. These comparisons are stored in two ways in POSTPROCESSING/COMPARE:

a. Each pair of runs is saved in a file called `Comparison-TIMESTAMP1-TIMESTAMP2.txt` where `TIMESTAMP1` and `TIMESTAMP2` are the `TIMESTAMPS` of the respective runs. If only one member of a file is identified, `TIMESTAMP2` is identical to `TIMESTAMP1`.

b. Each of the comparison files identified in the previous paragraph is concatenated into a single file, separated by a line indicating page breaks. The header of this file is “Appendix. Run-by-run comparison”. The file is named `Appendix-RunByRunComparisons-DATESTAMP.txt` where `DATESTAMP` has the format `YYYY-MM-DD` (year-month-date). This file can be opened with a text editor, then copied / pasted into Word. Change the font to 9-point Courier and replace the text “`PAGEBREAK`” with manual page breaks using Word’s “REPLACE” function (look for “Manual Page Break” in the Special tab).

Next, a table is created that includes the following columns:

1. Run number: This is the time stamp of the “new” model
2. Compare to model: This is the row number of the “old” model
3. Compare to run #: This is the time stamp of the “old” model
4. Additional parameters: This is the number of additional parameters in the “new” model compared to the “old” model (*i.e.*, the decrease in the number of the degrees of freedom)
5. Objective function of “new” model
6. The number of significant digits in the “new” model.
7. Were standard errors obtained (yes / no)?
8. Decrease in objective function: Objective function of “old” model – objective function of “new” model
9. The *P* value for likelihood ratio test comparing “new” model to “old” model (based on decrease in objective function and change in degrees of freedom, assumed nested models)
10. Number of non-fixed *thetas* in the “new” model
11. Number of non-fixed *omegas* in the “new” model
12. Number of non-fixed *sigmas* in the “new model
13. Comments entered in the Run Listing Editor.

Finally, the brief summaries for each run are concatenated into a single file (in the order indicated in the first column of the table in the graphical interface). The header of this file is “Appendix. Brief Summary of each NONMEM run”. The file is named `Appendix-BriefSummaries-DATESTAMP.txt` where `DATESTAMP` has the format `YYYY-MM-DD` (year-month-date). This file can be opened with a text editor, then copied /

pasted into Word. Change the font to 9-point Courier and replace the text

“ _____ PAGEBREAK _____ ”

with manual page breaks using Word's "REPLACE" function (look for "Manual Page Break" in the Special tab).

NOTE: Runs available for comparison are those at the Project level, rather than the Working Folder level. Thus, if the user maintains more than one Working Folder (*e.g.*, Working1 and Working2) within a Project Folder, runs generated from either Working Folder can be compared.

To use the Summarize / Compare tool, the user must identify the Project Folder (this need not be the same Project Folder as in the Project Controller window), then, populate the table. The first two columns indicate the "new" run and the "old" run, respectively. The "old" run is the run to which the "new" run should be compared. These cells are populated with time-stamps using pull-down menus. All runs should appear in these pull-down menus. The third column indicates a user-determined increase in the number of parameters in the "new" run compared to the "old" run. For example, addition of a single covariate to a model typically increases the number of parameters by +1; deletion of a covariate typically increases the number by -1 (value < 0 should be entered to indicate a decrease in the number of parameters in the new model. The fourth column allows the users to identify comments to be inserted into the table. Unlike the comments that can be embedded in a NONMEM Control Stream (which, if preceded by ";c" and appear on a separate line), these comments are restricted to 80 characters.

Once the table is populated, click the **Summarize / Compare** button. The time-stamped output will be created. Occasionally, error conditions may be encountered; these are sent to the Output window. If the message is that files relevant to a particular time-stamp cannot be located, it is the users responsibility to change the entries in the Run Listing (*e.g.*, if the user deletes files from any subfolder within TEXTFILES or TABLES, these cannot be accessed).

NOTE: The user may encounter other error messages, possibly due to unanticipated text in NONMEM output. If this occurs, please copy the error message and email it to support@PLTsoft.com so that we can change the algorithm.

The process of creating new comparisons can be repeated as necessary.

Assemble Report

The Assemble Report tool generates the same output as the Summarize / Compare tool, then it creates a rich-text (RTF) document that can be opened in Microsoft Word or Open Office (and edited, as appropriate). Elements in the document can be controlled by the user. These include:

Elements Accessed from All Runs

The following elements are accessed from all runs in a Run Listing (or from a single run if "Source" is "Single Run").

Brief Summaries, Run-By-Run Comparisons, Summary Table: These items are described in the previous section.

Covariate Graphics. Covariate Summaries: The first time that a dataset is analyzed (and any time that the data or a covariates file is changed), covariates are summarized in a table and in graphics.

Elements Accessed from Selected Runs

The user controls the runs from which the elements below are accessed. If Single Run is selected, elements from that run are selected. If Run Listing is accessed, the user can then select from "All" and "Selected Runs" in the "If Run Listing" pull-down menu. The "All" is selected, then elements below are accessed for all entries in the Run Listing. If "Selected Runs" is selected, then the user can select combinations of "Final", "Base", and "Other". A typical approach for a final report might be for the Run Listing to include all runs critical to describing the project, with one run identified in the Run Listing as Base (*e.g.*, a model with no covariates) and one model (the "optimal" model) identified in the Run Listing as Final. The user would then select "Run Listing" as Source, "Selected Runs" for "If Run Listing" and check "FINAL" and "BASE" (**Figure 29**).

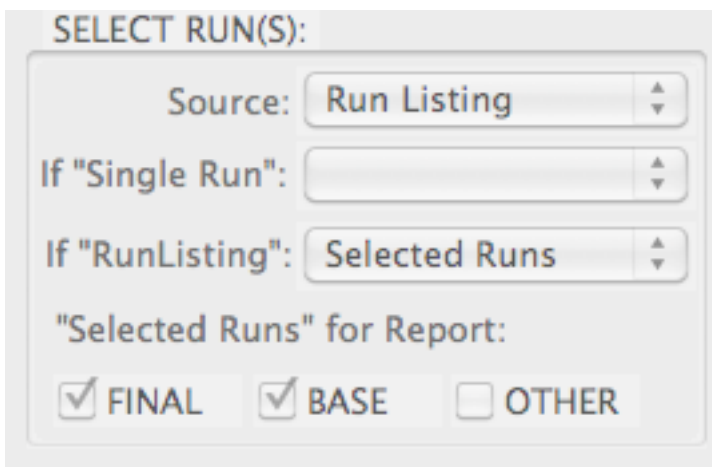


Figure 30. Sample Configuration for "Select Runs" in the Summarize / Report Tool.

The configuration shown here enables preparation of a report in which certain elements (*e.g.*, the Summary Table, Brief Summaries, and Run-by-Run Comparisons) include all runs in a Run Listing but parameter values, NONMEM files, and single-panel graphics are accessed only for the FINAL and Base runs.

Elements are:

NONMEM Control Stream, Output Files: The NONMEM Control Stream and/or the output file for each run # (timestamp) identified by the user.

Population Parameter Estimates, *Post Hoc* Parameter Estimates: These values are displayed in tables.

Single-panel graphics: At the completion of each NONMEM run, a set of graphics is created in JPEG format (ready to be embedded into other documents). These graphics are identical to those in the PDF graphics document except that they are sized more appropriately for insertion into a summary document. The user can select:

1. Observed vs. predicted. There are separate panels for population and *post hoc* fits.

2. Observed / predicted vs. time. There are separate panels for population and *post hoc* fits.
3. *Post hoc* ETAs vs. covariates.
4. *Post hoc* parameter estimates vs. covariates.

Single Panel Graphics: These graphics are created during the individual runs and are archived in the folder GRAPHICS-SINGLE.PANELS in subfolders by timestamp. Each graphic is named by the page in the combined PDF and type of graphic; each graphic is accompanied by a text file with the same name and a .txt extension.

Additional information regarding tailoring of this document (e.g., creating a customized cover page) is provided in the document itself.

Using the Archive Manager Tool

When a project is complete (or at any earlier time), the user can bundle files from relevant runs into an archive. This archive serves two purposes: only relevant runs are included and back-up is facilitated. The Archive Manager tool can perform its tasks on either a single run or from a Run Listing. To prepare a Run Listing, see "Using the Run Listing Editor"; using the pull-down menus and checkboxes, the user can then select to use all of the Run Listing or a subset (the BASE, FINAL, or OTHER runs or any combination thereof). Alternatively, the user can select a single run # (timestamp) by selecting "Selected Run" from the Source pull-down, then selecting the timestamp from the pull-down labeled 'If "Selected Run"'.

Next, the user selects the types of files to be archived (**Figure 30**) by checking the appropriate boxes in the "File Types to Archive" section of the Archive Manager tab. By default, all elements except "Extra Records" are checked.

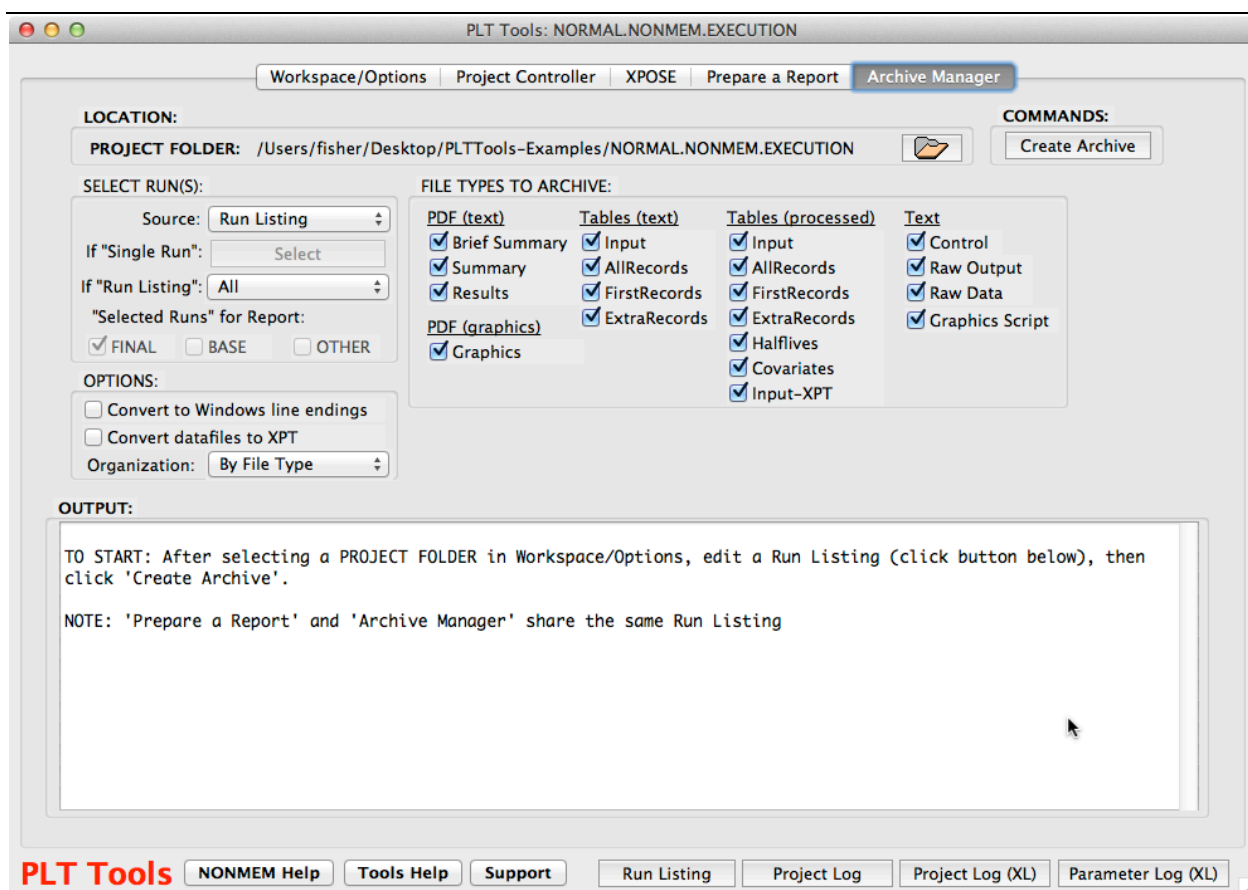


Figure 31. The Archive Manager tool.

Three options are available. The first of these, Convert to Windows line endings, applies only to analyses conducted in OS X and can be ignored for analyses performed exclusively in Windows. Linux, OS X uses different line endings than Windows. As a result, text file created in OS X cannot be read correctly by certain Windows applications (e.g., Notepad opens the entire file as a single line whereas Wordpad and Word open the file correctly). **PLT Tools** offers the option

"Convert to Windows line endings" to overcome this problem. If this option is selected, all text files added to the archive (files ending in .txt, .csv, and .tab) are converted so the line endings consistent with Windows' standards.

The second option "Convert datafiles to XPT" permits **PLT Tools** to convert datafiles to SAS XPORT (.xpt) format. This option can supplement or replace a similar procedure in which the FDATA file for each run is saved in the SAS XPORT format. However, this option refers to datafiles, rather than FDATA files. A major difference between datafiles and FDATA occurs only if the "drop" or "skip" options are applied in the \$INPUT record in a Control Stream. In addition, the FDATA file is saved with the column names assigned in the \$INPUT record whereas column names in datafiles may have been entered by the user and need not follow NONMEM conventions (e.g., NONMEM 5 and 6 restricted the length of columnnames to four characters).

The "Convert datafiles to XPT format works as follows:

1. The DATAFILES folder (and any subfolders) in the Project Folder are searched for any files with the extensions .txt or .csv. All other files are ignored.
2. Each of these files is examined by R to determine if the file is compatible with a datafile and can be read into R as a table. First, all rows beginning with a character other than a comma, a dot, or an integer are deleted; R then attempts to read the file assuming either a comma, a space, or a tab as the delimiter (depending on the name of the file). If an error condition does not occur, R then examines the previously-deleted lines. If there is only one, R assumes that this line contains the column headers. If there are none, column headers are ignored. If there is more than one, the first and last of the previously-deleted lines (restricted to lines preceding the first non-deleted line) are each examined to determine which can be split into the appropriate number of elements (e.g., if the data appear to contain 9 columns, the text lines are examined to determine which contains 9 elements). If neither line contains the appropriate number of elements, column headers are ignored. If both lines contain the appropriate number of headers, the first is assumed to contain the column headers. If only one line contains the appropriate number of headers, it is assumed to contain the column headers. The column names have not been obtained, the text "Column01", "Column02", is used instead.

The table is assigned these column headers. The table is then written to a file in SAS XPT format. In addition, the column names are written to a separate file whose name is modified by insertion of the text "-ColumnNames". All files contain the original filenames so that they can their origin can be traced. For example, a datafile named Data.txt would yield two files:

- a. SAS XPT file: Data.txt.xpt
- b. Column name file: Data.txt-ColumnNames.txt

The column name file contains one entry per row. The user can use this file to create a Define.pdf file, as is commonly required for a regulatory submission.

The third option, Organization, controls the file structure of the archive. The default, "By File Type", creates a different folder for each of the "Components to Archive" (if a particular component is not selected, no folder is created). The alternative, "By TIMESTAMP", creates a

separate folder for each timestamp (run #). If the user selects "By File Type" and if BASE, OTHER, and/or FINAL has been selected in the "Special" column in the Run Listing, additional folders are created. These folders are named based on the "Special" label and the timestamp (e.g., 0911190-091007-FINAL) and contain an additional set of copies of all files for that run #.

Note: Archive Manager *copies* files. Original files are never deleted.

To facilitate assembly of the Run Listing, the user can click [View Run Log](#) or [View Project Log](#). The run log consists of three columns. The first is the time stamp associated with every run conducted in the Project Folder. The second column is the same of the Control Stream that the user selected for that run. This is useful if the user names each Control Stream in an informative manner. The third column is the contents of the \$PROBLEM record from the Control Stream. This is useful if the user updates the \$PROBLEM statement in an informative manner. If the user always works from the same Control Stream and does not update the \$PROBLEM statement, the Run Log is not useful. The project log contains the same information as the run log plus the value of the objective function, the number of significant digits and other useful information.

When the [Create Archive](#) button is clicked, a new archive is created. First, the "New Model" and "Old Model" columns of the Run Listing are merged and duplicate values are deleted. Next, the requested files are copied to POSTPROCESSING/ARCHIVE in a new folder named ARCHIVE-TIMESTAMP, where TIMESTAMP has the same format as the individual files but indicates the time at which the archive was created. Thus, the user is able to repeatedly archive files during the course of a project.

If a particular file cannot be located (either because it was never created, e.g., if the user did not request calculation of half-lives, or it was moved or deleted by the user), a message will be sent to the Output window. If no files relevant to a particular time-stamp are found, a different message will be sent to the Output window.

Using the Run Listing Editor

The Prepare Report tool and the Archive Manager tool both may require that the user create a list of NONMEM runs. This is accomplished using the Run Listing Editor (**Figure 31**).

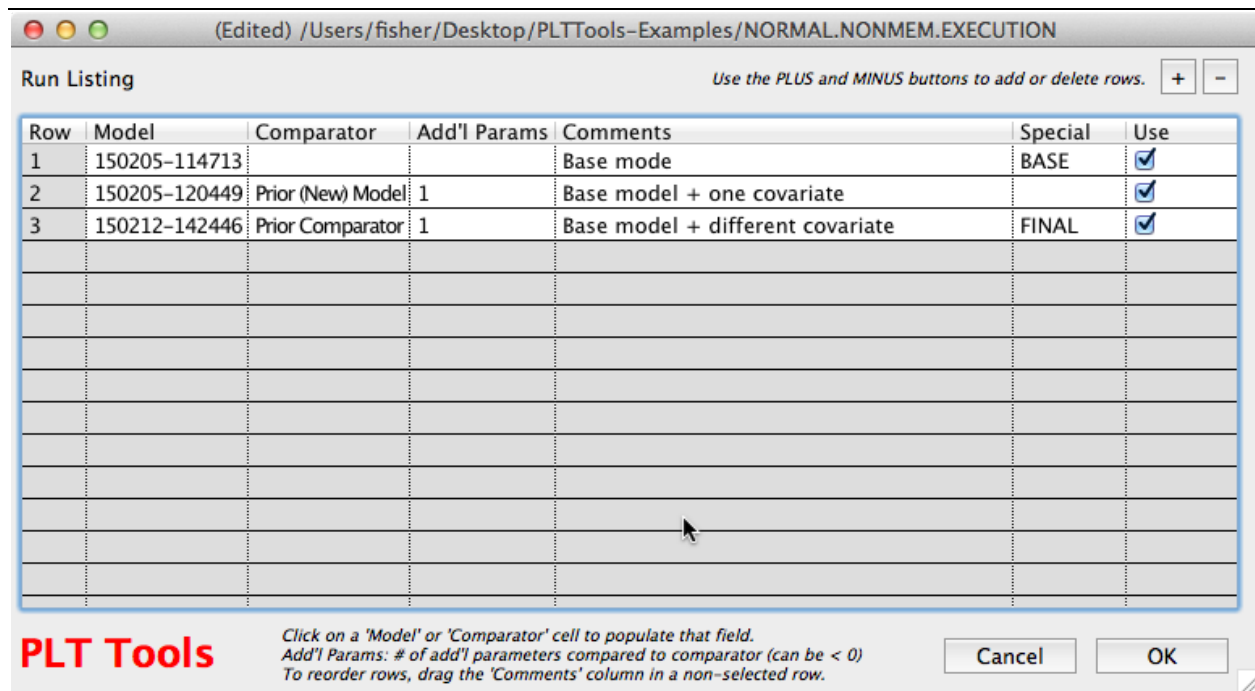


Figure 32. The Run Listing Editor is shown.

The Run Listing Editor is accessed by clicking the **Run Listing** button at the bottom of the window. If a Project Folder has not been selected, the user will be directed to select a Project Folder first.

The Run Listing Editor contains five columns (plus integers indicating row numbers). These columns are best explained by example in which the data analyst has performed a large number of runs, then, selected a subset of these to present in a report. The time-stamp for the first of these runs is entered in row 1 as the “Model” using the pull-down menu in that field. There is no “Comparator” for comparison so columns Comparator and Params are ignored. The user can enter descriptive text in the Comments field (*e.g.*, “Base model”). The fifth column allows the user to (optionally) identify a run as BASE, OTHER, or FINAL. The sixth column contains check boxes. If a row is unchecked, that entry is ignored when the Run Listing is invoked. These designations permit the user to create summaries, reports, or archives based on subsets of the Run Listing rather than the entire listing.


Next (row 2), the analyst the next model of interest. The time-stamp for this run is entered in row 2 as the “Model”. The time-stamp from the previous run is entered as the “Comparator” (*i.e.*, the analyst wants to compare the “new” model to the “old” model). The analyst must determine how many additional parameters there are in the new model; this value is entered in the Params column. Descriptive text can be entered in Comments column. Additional rows can be entered in the same manner.

Entering values for “Model” and “Comparator”: Click on any white cell in the columns with those names (rows that are gray are not available for entry; to make them available for entry, see “Adding Rows” below). A listing of the time-stamps of all NONMEM runs in that Project Folder will appear. Select the time-stamp of interest. If the list is lengthy, use the Search field to find the run of interest. The search field examines the contents of all columns.

Two specialized selections are available: “Prior (New) Model” and “Prior Comparator”. If the user selects either of these, during processing **PLT Tools** will replace these with the appropriate entry from the row immediately above (these values should never be selected for the first row of a run listing). If the user has previously entered a timestamp and wishes to clear the field, click “No Model”.

Entering values for “Addl’ Params” (additional parameters): If two models are to be compared using the likelihood ratio test (LRT), the user must enter the number of additional parameters in the “Model” compared to the “Comparator”. For example, if the new model adds a single parameter to model a covariate, a value of 1 should be entered. Conversely, if a new model deletes a single parameter to remove the effect of a covariate, a value of -1 should be entered. The user is responsible for determining whether the models can be compared with the LRT (*i.e.*, the models are nested) and for determining the number of additional parameters (*i.e.*, the decrease in the degrees of freedom between models).

Entering text for “Comments”: The user can enter up to 80 characters of text. This text will appear in the Comments field of the table created by the Compare Runs tool.

Adding rows: The Run Listing Editor opens initially with a single row that can be populated. To add additional rows, click the  button in the upper right corner.

Deleting rows: Rows can be deleted by clicking the  button in the upper right corner.

Re-ordering rows: Rows can be re-ordered by dragging a non-selected row.

xpose

xpose (xpose.sourceforge.net) is an R-based model-building aid for NONMEM developed and maintained by Hooker, Wilkins, Karlsson and Jonsson of Uppsala University. xpose creates graphics and tables interactively and independently of completion / filing of a NONMEM run. This contrasts to **PLT Tools**, which creates these documents immediately after a NONMEM run completes (although **PLT Tools'** graphics can be revised later).

Support for xpose was added to **PLT Tools** in Version 3.5 and updated in Version 5.1. These initial versions may not satisfy the needs of all users. As a result, we appreciate input as to which additional functions should be supported and any menu items / options that should be added. Contact us at support@PLTsoft.com (or click the Support button in **PLT Tools**).

Generating xpose Graphics and Tables: **PLT Tools** supports xpose as a supplement to its existing components. To generate xpose graphics and/or tables:

1. Click the **XPOSE** tab in **PLT Tools**.

STEP 1:

1. Click **Select** next to "Run:", then select a Timestamp.
2. Optional: Click **Select** next to "Comparator:", then Select a Timestamp

STEP 2:

1. Optional: Select "Input". The default is "PLT Tools pull-down", *i.e.*, using the pull-down menu labeled "STEP 4: XPOSE Command:". Users wishing to access commands not in that pull-down menu will need to select either "Local File" or "Remote File", then prepare a user-supplied script (see below).
2. Optional: Select a value for Rows/Columns (this applies only to By-Subject graphics).

STEP 3:

1. Click **Start**. **PLT Tools** will copy the appropriate files to WORKINGFOLDER/XPOSE-WORKING. These files will be renamed using the xpose naming convention (*e.g.*, sdtabFILEID, patabFILEID) where FILEID is the 13-character timestamp. If **PLT Tools** is unable to find appropriate files, messages will appear in the Output window.

STEP 4:

1. Using the **XPOSE Command** pull-down menu, select the desired graphic or table. The names that are displayed are similar to the corresponding xpose commands (mapping of **PLT Tools** versions to actual xpose commands is displayed in **Table 2, Table 3**).

2. Click **Submit**. **PLT Tools** will attempt to execute the R function. If a PDF graphics does not open shortly thereafter, check the Output window for messages indicating possible reasons for failure. The selection can then be changed and a new graphic generated.

6. When the session is complete, click **End**.

Table 2. xpose Graphics Functions and the Corresponding Menu Items in **PLT Tools**.

xpose Function	Corresponding Item in PLT Tools Pull-Down Menu
<code>dv.vs.idv()</code>	COMPOSITE DV VS. TIME
<code>dv.preds.vs.idv()</code>	COMPOSITE DV/PRED/IPRED VS. TIME
<code>ind.plots()</code>	INDIVIDUAL PLOTS
<code>dv.vs.pred.ipred()</code>	DV VS. PREDICTIONS
<code>parm.hist()</code>	PARAMETER HISTOGRAM
<code>parm.qq()</code>	PARAMETER QQ PLOT
<code>parm.splom()</code>	PARAMETER SCATTERPLOT
<code>ranpar.hist()</code>	ETA HISTOGRAM
<code>ranpar.qq()</code>	ETA QQ PLOT
<code>ranpar.splom()</code>	ETA SCATTERPLOT
<code>parm.vs.parm()</code>	PARAMETER VS. PARAMETER
<code>parm.vs.cov()</code>	PARAMETER VS. COVARIATES
<code>cov.hist()</code>	COVARIATE HISTOGRAM
<code>cov.qq()</code>	COVARIATE QQ
<code>cov.splom()</code>	COVARIATE SCATTERPLOT
<code>absval.cwres.vs.cov.bw()</code>	ABSOLUTE WRES VS. PRED (box-whiskers)
<code>absval.iwres.vs.ipred()</code>	ABSOLUTE IWRES VS. IPRED
<code>absval.iwres.vs.pred()</code>	ABSOLUTE IWRES VS. PRED
<code>absval.cwres.vs.ipred()</code>	ABSOLUTE CWRES VS. IPRED
<code>cwres.vs.pred()</code>	CWRES VS. PRED
<code>absval.wres.vs.cov.bw()</code>	ABSOLUTE WRES VS. COVARIATES (box-whiskers)
<code>absval.cwres.vs.cov.bw()</code>	ABSOLUTE CWRES VS. COVARIATES (box-whiskers)
<code>cwres.dist.hist()</code>	CWRES HISTOGRAM
<code>cwres.dist.qq()</code>	CWRES QQ

Table 3. xpose Table / Text Functions and the Corresponding Menu Items in **PLT Tools**.

xpose Function	Corresponding Item in PLT Tools Pull-Down Menu
<code>parm.summary()</code>	PARAMETER SUMMMARY (table)
<code>cov.summary()</code>	COVARIATE SUMMARY (table)
<code>tabulate.parameters()</code>	TABULATE PARAMETERS (text)

User-supplied Script: The user can prepare a file (XPOSE-COMMANDS.R) containing a series of xpose commands. This permits access to a variety of functions not listed above. These functions typically refer to a “database”, an object created by R. The two relevant objects are MODEL.DB and COMPARATOR.DB (both are which are created by **PLT Tools** from files identified above). For example, the file could contain the text:

```
print(ind.plots(MODEL.DB))
```

This would yield graphics identical to selecting the "INDIVIDUAL PLOTS" item in **PLT Tools**' menu.

If the file contains:

```
cov.summary(MODEL.DB, out.file="covar")
```

the covariate summary table is created.

These commands must be placed in a text file named XPOSE-COMMANDS.R (do NOT append a .txt extension). This file can be located in either of two locations:

PROJECTFOLDER/XPOSE-TEMPFOLDER: If the file is located here, it applies only to this PROJECTFOLDER.

PLTTools-Support: If the file is located here, it applies to all projects. This folder can be accessed from the File -> Open menu.

When **Start** is clicked, **PLT Tools** searches for script files. If located, they are executed immediately. Graphics are created and filed automatically.

The file should **NOT** contain extraneous R commands such as "cat", "sink", "pdf", "write", "dev.off", or "graphics.off"; files containing these commands will not be executed.

If the file contains commands to create graphics, remember to pre-pend the "print" or "plot" commands; objects created by xpose are not printed automatically.

If padding (space around the panels) is not correct in user-supplied graphics, you need to provide instructions to xpose. For example:

```
print(absval.wres.vs.cov.bw(MODEL.DB, par.settings = PADDING))
```

applies the default 'PADDING' settings. Users experienced in the use of R's lattice graphics can apply their own options.

Certain xpose functions (e.g., `parm.summary`) output a table. The default behavior is for this table is sent to the "screen". However, this option is not available in **PLT Tools**. Instead, the user needs to direct the output to the appropriate file. For example, the command `parm.summary` should be formatted:

```
parm.summary(MODEL.DB, out.file="filename.txt")
```

Other xpose functions require `outfile` rather than `out.file`; the user is responsible for determining the correct syntax.

Files Required by xpose: xpose searches for files named `sdtabn`, `patabn`, `cotabn`, and `catabn` (and others), where "n" is an integer. There are two methods to create files for use by xpose:

1. Using `AllRecords` (and `FirstRecords` and a `Covariates` file, if appropriate) and a `Graphics Script`: If the user creates an `AllRecords` file (see elsewhere in this manual), **PLT Tools** assembles the file `sdtab` in xpose format (and a copy in csv format) automatically;

however, if the user has created a file named sdtab, **PLT Tools** takes no action (except to rename that file – see below).

If the user creates a Graphics Script, up to four xpose files are created with each NONMEM run (**Table 3**); these files are assembled from AllRecords and, if they exist, FirstRecords and/or the Covariates file. If other xpose files are needed, use the approach below to create them.

Table 3. xpose Tables Created by **PLT Tools** .

Name Expected by xpose	Data Items (Columns in AllRecords, FirstRecords, Covariates File)	Names Assigned by PLT Tools *
sdtab†	ID, TIME, DV, PRED, RES, WRES, IWRES, IPRED, IPRE, IPRD, CWRES, EPRED, ERES	SDTAB.TIMESTAMP.txt SDTAB.TIMESTAMP.csv
patab†	ID, THETA and ETAS listed in Graphics Script	PATAB.TIMESTAMP.txt PATAB.TIMESTAMP.csv
catab†	ID, all categorical covariates listed in Graphics Script	CATAB.TIMESTAMP.txt CATAB.TIMESTAMP.csv
cotab†	ID, all continuous covariates listed in Graphics Script	COTAB.TIMESTAMP.txt COTAB.TIMESTAMP.csv

* TIMESTAMP refers to the 13-character timestamp used to name all files in **PLT Tools** .

† Note that the integer that usually follows the xpose filename is omitted.

NONMEM allows aliases, e.g., CP=DV, in the Control Stream. These aliases are used by NONMEM in all tables. If aliases are used, **PLT Tools** anti-aliases these names.

2. As an alternative, the user can add \$TABLE statements to the Control Stream. The entry should specify FILE=FILENAME, where filename can be any of the xpose files, sdtab, patab, catab, mutab, mytab, xptab. However, an integer should NOT be appended to the filename (e.g., sdtab, not sdtab1). If **PLT Tools** finds a file with any of these names, it will file it in PROJECTFOLDER/XPOSE-TABLES after it is renamed using the **PLT Tools** naming conventions; in addition, a csv version will be created.

The user can mix these two approaches.

Graphics: All graphics created in xpose by **PLT Tools** are filed in PROJECTFOLDER/XPOSE-GRAPHICS in a subfolder named after the 13-character timestamp. In addition, the graphic is copied to the folder PROJECTFOLDER/XPOSE-WORKING/GRAPHICS.

Obtaining xpose software: If the user has not previously installed xpose (an R package), **PLT Tools** will inform the user. xpose4 can then be downloaded through the following steps:

1. Open the R graphical interface.
2. In the Package menu, select “Package Installer”.

3. Search for “xpose4”. Several packages (xpose4, xpose4classic, xpose4data, xpose4specific; the present version is 4.5.0) should appear; *all are necessary*. Install these packages and any dependencies that are required.

4. In R, type:

```
require("xpose4")
```

xpose should now be available.

5. In R, type:

```
xpose4()
```

If R responds

```
Error: could not find function "xpose4"
```

the installation was not successful. A successful installation will return a number of messages such as:

```
Loading required package: xpose4"
```

6. Once xpose4 has been installed successfully, **PLT Tools** will be able to create xpose graphics and tables.

Limitations: NONMEM analyses conducted in **PLT Tools** versions prior to 3.5 did not create and/or file xpose tables. Attempting to run xpose on these earlier runs will yield a message: “ERROR: **PLT Tools** did not identify any XPOSE files”.

Obtaining Help

Help can be obtained for each of **PLT Tools** and NONMEM.

Obtaining Help for PLT Tools

Help for **PLT Tools** is available *via* two routes:

1. Click the **Tools Help** button on the lower border of the graphical interface (**Figure 32**). This document will open.

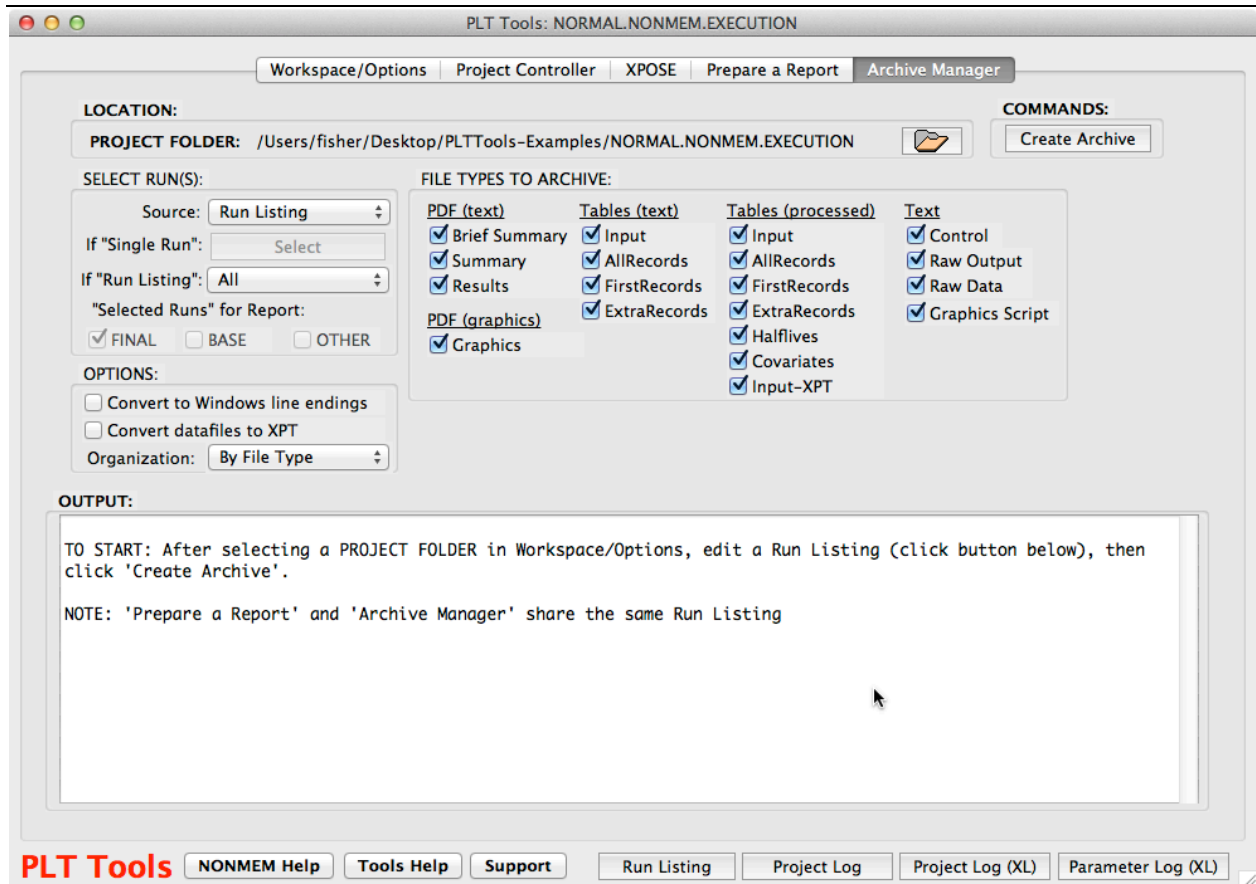


Figure 33. The lower border of **PLT Tools** contains two buttons to assist the user with **PLT Tools**.

2. Click the **Support** button on the lower border of the graphical interface. A message box will appear (**Figure 33**). After the user clicks OK, an email, addressed to support@PLTsoft.com will appear and a folder containing certain files useful for debugging will open. Describe the exact problem and, if it relates to a particular NONMEM run, include (if possible) the Control Stream and the data. If the problem relates to graphics, also include the Graphics Script. Finally, the folder, `DEBUGGING.FOLDER` (which can be accessed from the menus:

FILE -> OPEN -> Preferences (AppData))

contains several files that are useful to debug potential problems. Please attach these files to the email.

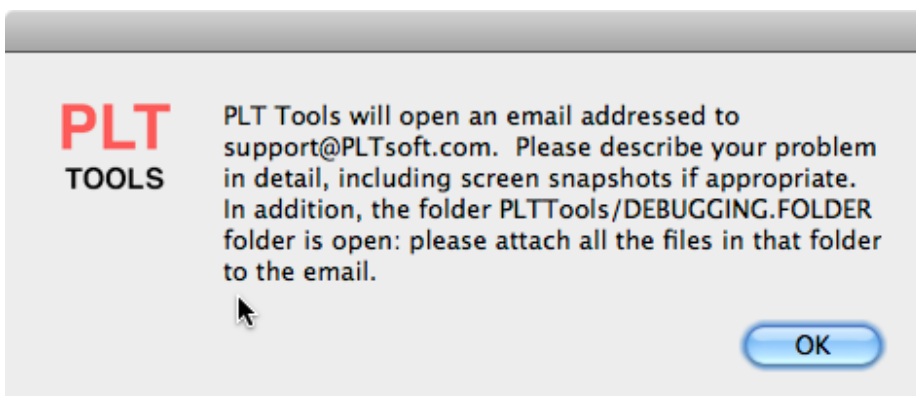


Figure 34. If the user clicks **Support**, this message appears.

Obtaining Help for NONMEM

Each NONMEM distribution contains a series of help files in a folder “help” (e.g., if NONMEM is installed in C:\nm7, help files are located in C:\nm7\help). To access these files from **PLT Tools**, click the NONMEM Help button. A window opens (**Figure 34**). Enter a search term and the initial list will display only the relevant items. When the user selects one item, the corresponding “help” document appears. The document can be printed.

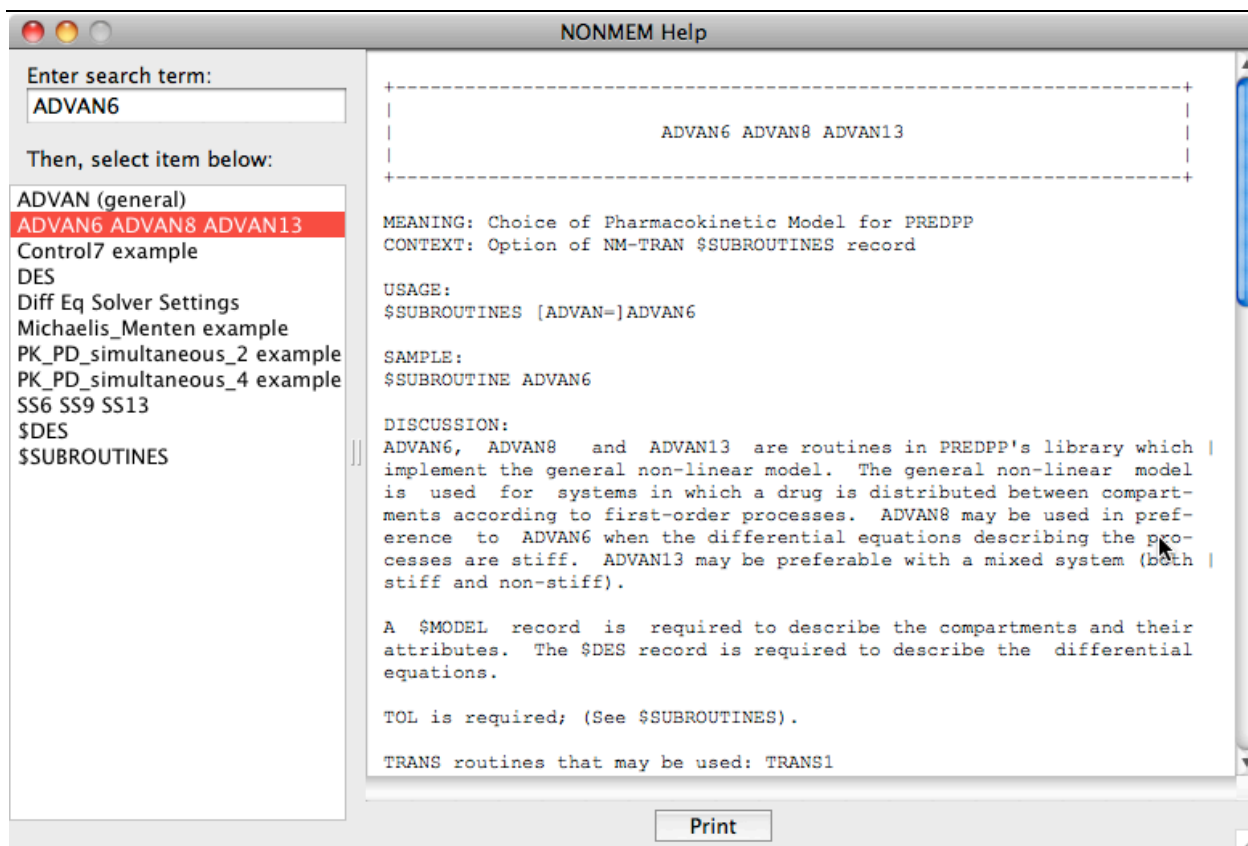


Figure 35. If the user clicks **NONMEM Help**, NONMEM’s help files (provided as part of the NONMEM installation) can be accessed and printed.

The first time that the user requests help, it may be necessary to provide **PLT Tools** with the /path/to/nonmem/help.

Assumptions Made by PLT Tools

The general approach to the development of **PLT Tools** was to make as few as possible assumptions regarding NONMEM input and output. However, a few assumptions have been made regarding headers of output tables. Headers are interpreted as follows:

Term appearing in NONMEM tables	Assumed Meaning
IPRE, IPRD, IPRED	<i>post hoc</i> predictions
IRES	<i>post hoc</i> residuals
IWRE, IWRES	<i>post hoc</i> weighted residuals.

Note that NONMEM 5 and 6 truncate names longer than 4 characters; thus, \$TABLE instructions requesting inclusions of IPRED result in IPRE appearing as the column header.

Certain information that the user does not enter into tables is retrieved by **PLT Tools** from other sources. For example, if the user does not enter MDV into a table, it is extracted from the FDATA table created by NONMEM. Messages are sent to the Project Controller Output window indicating that this has been done.

Troubleshooting

Once installation is successful, **PLT Tools** typically works consistently without errors. However, there may be problems related to installation. Installation can be confirmed with the following steps:

Windows:

1. Open a Command Prompt window.* Do not change any environment variables.
2. Navigate to the folder in which your test of **PLT Tools** failed.
3. type: `\Program Files\PLTTools\nmfe6.bat CONTROLSTREAM NONMEMOUTPUT` where CONTROLSTREAM is the name of a Control Stream in that folder. If **PLT Tools** runs successfully, the installation is correct. Otherwise, some component of the installation is flawed.

OS X:

1. Open a terminal window. Do not change any environment variables.
2. Navigate to the folder in which your test of **PLT Tools** failed.
3. type: `/Applications/PLTTools/nmfe6 CONTROLSTREAM NONMEMOUTPUT` where CONTROLSTREAM is the name of a Control Stream in that folder. If **PLT Tools** runs successfully, the installation is correct. Otherwise, some component of the installation is flawed.

Once you have evaluated the installation of NONMEM, evaluate which, if any, of the messages below appeared. Common messages include:

Message: Unable to get Tool version: **PLT Tools** calls R as the internal engine. This error message indicates that an attempt to call R failed. Possible explanations:

1. The path to R is set incorrectly in Preferences.
2. In Windows, the user selected an R executable that was not RTerm.exe (e.g., R.exe or RGUI.exe).

If fixing these problems does not resolve the issues, contact support@PLTsoft.com.

Message: This suggests that the original installation of Fortran is flawed: Before **PLT Tools** runs NONMEM, it examines the file nmfe6.bat to identify the Fortran compiler that will be called during NONMEM runs. **PLT Tools** then sends a null command† to confirm that the user's environment can find this Fortran compiler. If the operating system does not receive the appropriate response, this message is sent; the message indicates that the /PATH/TO/Fortran is not known to the operating system. This can be confirmed by opening a Command Prompt (in OS X, a Terminal Window), then type the command, e.g., "G95". The operating system should respond "No input files". If the response is similar to "G95 was not recognized", then Fortran is not known to the operating system.

* In the Windows Start menu, type "cmd" in the Search field, then click "Return". A command prompt window should open.

† For example, if the compiler is G95, the command is: `> G95`.

The solution is to update the environment variables. In Windows, this is accomplished in the "User Accounts" Control Panel; if you are not familiar with this procedure, see <http://support.microsoft.com/kb/310519>. The procedure in OS X is described in http://www.tech-recipes.com/rx/2621/os_x_change_path_environment_variable.

After the environment variables are fixed, open a new Command Prompt or Terminal Window, then type the command again. If the operating system still cannot find the Fortran installation, contact support@PLTsoft.com.

Message: NONMEM has not started: This text may appear for an extended period during the first use of **PLT Tools**. Typically, it is caused by problems related to environment variables (see highlighted sections below). The problem can usually be solved by minor changes in the installation procedure. If the steps outlined below do not resolve the problem, contact support@PLTsoft.com for recommendations. In your email, include the file:

\Program Files\PLTTools\nmfe72.bat

(remove the .bat extension to ensure successful emailing) or

/Applications/PLTTools/nmfe72 (OS X)

If possible, also include your Control Stream and dataset.

gfortran in OS X: If gfortran is used, nmfe72 (in the PLTTools-Support folder) contains the code:

f=gfortran

When NONMEM is called from **PLT Tools**, the operating system may not be aware of the path to gfortran (which is typically /usr/local/bin). This can be fixed with the following steps:

1. In **PLT Tools**, FILE >- OPEN -> Support (Preferences) Folder.

2. Using a text editor (e.g., TextEdit), edit the file nmfe72.

3. Replace the line above with:

f=/usr/local/bin/gfortran

(if gfortran is in a different location, use the appropriate path; the path can be found by typing:

which gfortran

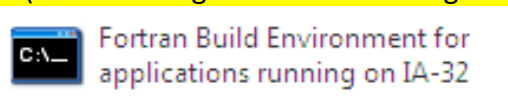
in a terminal window).

4. **PLT Tools** should now run successfully. If not, contact support@PLTsoft.com.

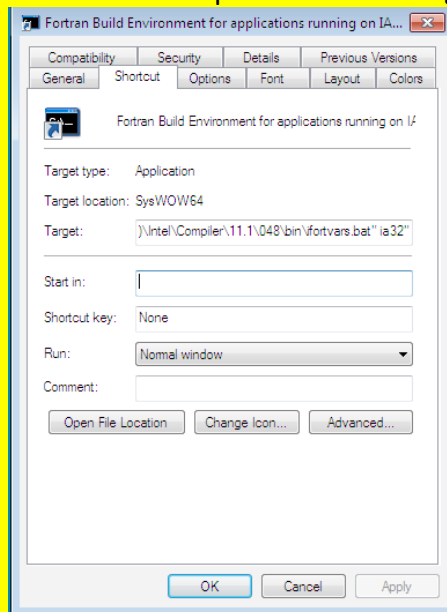
Intel Fortran in Windows: Intel's compiler makes extensive use of environment variables. These environment variables are set by a batch file named `ifortvars.bat`, provided with the Fortran installation. This information must be passed to **PLT Tools**. To accomplish this, follow either Approach 1 or Approach 2 below:

Approach 1:

1. Search Window's START menu for an item named "Fortran Build Environment for applications running on IA32" (or something similar — see image below).



If you find that item, right-click and select "Properties". The image below should appear.



The field labeled "Target" contains a lengthy path, *e.g.*:

`\C:\Windows\SysWOW64\cmd.exe /E:ON /V:ON /K ""C:\Program Files (x86)\Intel\Compiler\11.1\048\bin\ifortvars.bat" ia32"`

Ignore the first portion of this path (*e.g.*, `\C:\Windows\SysWOW64\cmd.exe /E:ON /V:ON /K`). The second portion is:

`""C:\Program Files (x86)\Intel\Compiler\11.1\048\bin\ifortvars.bat" ia32"`

This is the critical portion.

2. In **PLT Tools**, select the following menu items:
FILES -> OPEN -> Preferences (APPDATA) Folder
3. To edit `FortranVars.bat`, right-click on the file `FortranVars.bat` and select "Edit".
 - i. Identify any of the lines containing `@call`. Delete the text `rem` from **one** of these lines.
 - ii. In the **same** line, replace the path presently shown with the exact path identified in #1.
 - iii. Add the correct option, as determined in step #1. Follow the syntax shown in the file.
 - iv. In the final line of the file, delete the text `rem`.
 - v. Save the changes.
4. Run **PLT Tools** again. If **PLT Tools** does not run correctly, contact support@PLTsoft.com.

Approach 2:

1. Locate the file `ifortvars.bat`. Typically this file is in a folder in "Program Files" that contains the Intel Fortran compiler, e.g., "C:\Program Files\Intel\Compiler" or "C:\Program Files (x86)\Intel\Compiler".
2. Open a Command Prompt window* and navigate to the folder containing `ifortvars.bat`.
3. Determine which option (also known as a 'switch') `ifortvars.bat` requires. Typical options are: `ia32` (for a 32-bit installation) or `ia64` or `intel64` (for a 64-bit installation). It is possible that no option is required. To determine which option is correct, type:
`ifortvars.bat`
into the Command Prompt window, followed by a carriage return. If Windows replies:
ERROR: Unknown switch
or
The system cannot find the path specified.
try other options (e.g., type: `ifortvars.bat ia32`) until Windows replies with a message that starts with:
Intel(R) Visual Fortran Compiler ...
If you see this message, you have identified the correct option.
4. In **PLT Tools**, select the following menu items:
FILES -> OPEN -> Preferences (APPDATA) Folder
5. To edit `FortranVars.bat`, right-click on the file `FortranVars.bat` and select "Edit".
 - i. Identify any of the lines containing `@call`. Delete the text `rem` from one of these lines.
 - ii. In the **same** line, replace the path presently shown with the exact path identified in #1.
 - iii. Add the correct option, as determined in step #3. Follow the syntax shown in the file.
 - iv. In the final line of the file, delete the text `rem`.
 - v. Save the changes.
6. Run **PLT Tools** again. If **PLT Tools** does not run correctly, contact support@PLTsoft.com.

User Account Control (Windows): Microsoft introduced "User Account Control" (UAC) in Vista, a feature to prevent accidental installation of malware in Windows. UAC was then revised in Windows 7. If UAC is selected (which is the default for most installations), installation of **PLT Tools** becomes more complicated.

Installation in Program Files (Windows): UAC normally prevents installation in Program Files (or Program Files (x86) in a 64-bit environment). If the user has administrative privileges, it may be possible to install in Program Files (or Program Files (x86)) by right-clicking on the **PLT Tools** installer and selecting "Run as Administrator". However, this may create new problems:

- a. the folder `PLTTools-Support` (which contains Preferences and other critical files) is created in a location identified by the environment variable `%APPDATA%` (the path is something like `\Users\fisher\AppData\Roaming\PLTTools-Support`). This folder may be created by Windows

* In the Windows Start menu, type "cmd" in the Search field, then click "Return". A command prompt window should open.

without write privileges.

b. the folder PLTTools-Examples (located on the Desktop) may also be set up without write privileges.

If this occurs, **PLT Tools** may send error messages explaining the problem. Instructions are provided below as to how to change privileges.

Installation on the Desktop: This location is “safe” – the user should always have write privileges for the Desktop. However, **PLT Tools** may not be able to perform automation when the automation scripts are generated by **PLT Tools** (see User's Manual for more information). If you encounter this situation and wish to run automation scripts, please contact support@PLTsoft.com so that we can provide a solution.

Change Write Privileges in Windows: In Windows Explorer:

- a. Select the folder in which you cannot write.
- b. Right-click, select “Properties”.
- c. At the bottom of the window, you will see “Attributes”. “Read-only” may be checked (or the box filled in). Uncheck “Read-only”.
- d. It may be necessary to perform the same steps with the enclosing folder and all folders within this folder.

Contacting PLTsoft.com

Most problems can be resolved with a single email. This is most likely to occur if the user provides relevant information. When contacting support@PLTsoft.com, please do the following:

1. Click the Support button in **PLT Tools**. Follow the instructions.
2. If possible, obtain a screen snapshot of the error message.
3. If possible, send your Control Stream and dataset. If the problem relates to graphics, send the Graphics Script file.

Typically, you will receive a reply in less than one business day.

Acknowledgements

NONMEM would not exist without the efforts of our mentors Lewis Sheiner and Stuart Beal. We dedicate these efforts to them. NONMEM would be less useful without the efforts of Alison Boeckmann, the programmer for the NONMEM Project Group for 20 years. In particular, the nmsee code that she created is essential to our efforts. We thank Michael Diehr of Xochi Media and Joe Strout of Luminary Apps for their efforts to create and improve the graphical interface. Finally, we thank Zachary and Jeremy Fisher and Melanie Adams for inspiring better code.

Version History

Version 6.4.0. Released 2021-01-01.

New Features:

1. If 'Updated' is selected for Covariates for Graphics Only, text indicating that that updated covariates file was used can be suppressed.

Version 6.2.0. Released 2020-11-15.

New Features:

1. Support for NONMEM 7.5 added.

Version 6.1.0. Released 2020-05-22.

New Features:

1. In By-Subject graphics, an option exists to display the contents of a data item from the AllRecords file other than PRED/IPRED. This is accomplished with Cn (search this document for that string). Previously, the label on the y-axis was Cn (where n is a single digit). The option now exists to select a label for that axis.
2. Mail: Previously, constraints on the options for send mail from R required the user to install Java. These constraints no longer exist. The new approach requires the user to install an email package; details are provided in the body of this document.
3. NSAMP period marker for BySubject and Spaghetti graphics: If the multiple period marker for BySubject or Spaghetti graphics is followed by “|X.X”, where X.X is a real number, PLT Tools will stratify graphics for NSAMP. See highlighted text in the body of this document.

Version 6.0.1. Released 2018-02-25.

New features:

1. Tornado Plots: Tornado plots can be created once the user creates a project-level file containing instructions for tornado plots. These plots can be produced for every NONMEM run or during post-processing.
2. Parallel NONMEM runs; **PLT Tools** supports parallelization of NONMEM runs. The user must first configure NONMEM to execute parallel runs (see NONMEM guide nm742.pdf [or older versions]). Minor additional configuration is required within **PLT Tools**.

Updates to Existing Features:

1. The package used by **PLT Tools** to create SAS XPT files was outdated. A new version has been supplied with the installer.

Version 5.5.2. Released 2017-07-01.

Changes to existing features:

1. Single-panel graphics: Most (but not all) of the single-panel graphics are now created in both PDF and JPEG versions. The PDF versions embed better in Word documents. In addition, the file structure for these files is improved: each type of graphics (*e.g.*, covariate plots, observations vs. predictions) has been moved to a separate folder (*e.g.*, `COVARIATE.PLOTS`, `OBS.VS.PRED`) and within that folder, there are separate folders for JPEG and PDF graphics and captions (`TEXT`)
2. Likelihood profiles, VPC, Bootstrap: The single-page images, previously provided only in JPEG format, are now provided in both PDF and JPEG versions.
3. Smoothing records: Previously, if smoothing records were added to the dataset (`EXTRADATARECS`), time-varying covariate entries must have been out of sequence. A new feature fixes this if the user adds a record to the Control Stream and creates a text file instructing **PLT Tools** as to the relevant column names.
4. Likelihood Profiles: Previously, `NONMEM` runs were conducted in the sequence from the lower bound to the upper bound. In order to permit more rapid evaluation of appropriateness of these bounds, the most extreme values are evaluated first.

New features:

1. A file named `Define.TIMESTAMP.csv` is created for each run. The file is located in `TABLES/DEFINE.INPUT`. During creation of an `ARCHIVE` by the Archive Manager, that file is copied to a folder `DEFINE.INPUT`

Bug Fixes:

1. In `BySubject` graphics, under an obscure set of conditions, the x value of `IPRED vs. time` could be displaced laterally.
2. General Layout/Options tab: In the 'Select Graphics' section, if Residuals was not checked, the status of "Log-Log DV" and "Linear-Linear DV" was considered unchecked, regardless of actual status. This could result in the inability to select subsets of graphics.
3. BQL Graphics: If `NONMEM`'s `ADDL/II` feature was used and a subject had more than one explicit dose, Time After Dose was calculated incorrectly for second and later doses under certain circumstances.

Version 5.5.1. Released 3 January 2017.

Changes to existing features:

1. Time limit for by-subject graphics. If the user enters a negative value for the x-axis limit in the Graphics Script editor, PLT Tools applies the following approach:

- a. the absolute value of the entry is determined.
- b. the x-axis upper limit is calculated in the normal manner.
- c. if the usual upper limit is less than the entry, the entry value is applied
- d. if the usual upper limit is larger than the entry, the usual upper value is applied.

This approach allows the user to establish a minimum value for the x-axis upper limit but not truncate graphics for subjects with larger values. This feature has undergone limited testing – if problems are identified, please contact support@PLTsoft.com

2. Dose magnitude display in by-subject graphics. A new option was added – “First in sequence”. This is useful in the following situation: a subject has many repeated doses of the same magnitude, then a change to a new dose level. The “First Dose Only” option would not indicate that the dose changed whereas the “At All Doses” option might result in overlapping text (if there were a large number of doses). This new option minimizes overlap while presenting the critical data. This feature has undergone limited testing – if problems are identified, please contact support@PLTsoft.com

Version 5.5.0. Released 20 August 2016.

Changes to existing features: If there is a reset event (EVID = 3 or 4) and the user has not implemented the “within-subjects” feature correctly, PLT Tools attempts to determine “periods” by looking for decreases in the TIME variable between successive records within a subject. Graphics associated with this are now markedly improved and there is a new section in this manual explaining “reset events”.

Version 5.4.1. Released 17 February 2016.

Changes to existing features:

1. Extra data records: The algorithm is markedly improved.
2. Extra data records: Records are added relative to ADDL doses. This can be overridden.
3. Combining bootstrap runs or likelihood profile runs: The algorithm is markedly improved including better testing for duplicate records.

Version 5.2.0. Released 3 August 2014.

Changes to existing features:

1. Xpose: Updated to accommodate (and require) Xpose version 4.5.0. Improved layout for Xpose graphics. Minor bug fixes.
2. User-modified labels in By-Subject Graphics: Bug fixed.

Version 5.0.0. Released 30 August 2013.

Changes to existing features:

1. Bootstrap / Jackknife: When a (random) seed is not selected by the user, **PLT Tools** searches for a stored value (from a previous bootstrap / jackknife run). If none is found, 1 is applied. If a value is found, the stored value is incremented by 1. This assures that a different seed will be applied for each analysis.
2. Extra data records: Previously, **PLT Tools** was able to add extra data records (to permit smoothing of prediction vs. time curves. The user entered the desired number of records relative to each dose and an end-time relative to dose. This approach is still supported with one addition: a record is now added at the exact dose time. An additional approach allows the user to specify times for additional records.
3. Modification of BQL graphics: If there are any observations > LOQ that are not used in the analysis (EVID = 2 or MDV = 1), a graphic displays these values vs. the corresponding population and *post hoc* predictions.

New features:

1. Email notification: **PLT Tools** can send email notifying the user that a run is complete. In addition, it can provide periodic updates on the status of a bootstrap analysis or likelihood profile (e.g., every Nth run). A program to allow configuration of this feature is included with the installation.
2. Formatted “final” parameters: **PLT Tools** collects the final estimates for parameters (*thetas*, *omegas*, *sigmas*) and formats them in a manner suitable for entry in a Control Stream. This facilitates two scenarios:
 - a. The user wants to start a new run using the final estimates from a previous run.
 - b. The user wants to perform simulation using the parameters from a previous run.

In either case, edit the Control Stream using the file

`FormattedParameters.TIMESTAMP.txt` (in `MostRecentRun-Copies`; the original version of this file is located in `TABLES/FORMATTED.PARAMETERS`).

3. Skip base run: If the user is starting a bootstrap, likelihood profile, or jackknife analysis, the “base run” can be skipped. This is appropriate only if the base run is IDENTICAL to a previous run.

Version 4.6.5. Released 18 November 2012.

Changes to existing features:

1. Exponentiation of “take logs both sides”: If the “take logs of both sides” approach is invoked and there are records with EVID=2 or MDV=1, these DV values can be prevented from being displayed by entering DV as -9999.

2. VPC: Minor modifications to VPC graphic formats.

Version 4.6.2. Released 18 October 2012*New features:*

1. "Create time after dose": **PLT Tools** can add "time after dose" values automatically to AllRecords.
2. Suppress creation of sub-tables and individual graphics during simulation / VPC: If SUBPROBLEM is > 1, **PLT Tools** normally undertakes two lengthy tasks: creating individual tables for each subproblem (in addition to the master table) and generating graphics. Options now exist to prevent these steps.

Changes to existing features:

1. VPC: With large files (e.g., a large number of sub-problems), file manipulation was slow. The procedure was streamlined.
2. VPC: Format for VPC graphics was modified. The 90% confidence interval is now shaded. Previously, a smoother of DV values was displayed; this was replaced with lines at the 5th, 50th, and 95th percentile of DV values.

Version 4.6.1. Released 5 September 2012*New features:*

1. Full support for new NONMEM methods. The most visible change is a reorganization of the Brief Summary when more than one method is used in an analysis.
2. Options: MESSAGE tab (previously labeled DEBUG) allows user to suppress certain common messages. This list can be expanded with specific requests from users.

Version 4.6.0. Released 20 August 2012*New features:*

1. Bootstrap graphics now display parameter distributions as histograms (in addition to the cumulative distribution display previously implemented).
2. Support for Mountain Lion (OS 10.8, Apple) added.
3. Visual predictive check: If a Graphics Script contains covariate data, VPC and prediction-corrected VPC stratify for each categorical variable (if there are 4 or fewer groups) and continuous variable (stratified into 4 bins).

Changes to existing features:

1. Users now select a PROJECT FOLDER rather than a WORKING FOLDER. If the selection is potentially ambiguous (e.g., more than one potential WORKING FOLDER within a PROJECT FOLDER), a new dialog box allows the user to select the correct WORKING FOLDER.
2. Visual predictive check: If the datafile or the number of subproblems was very large, the large size of the AllRecords file slowed creation of graphics markedly. To address this, the procedure for making VPC graphics was revised markedly.

Version 4.5.0. Released 15 May 2012*New features:*

1. BQL Graphics: If any records in the dataset include $DV > 0$ and either $MDV = 1$ or $EVID = 2$, a graphic is prepared showing the population and *post hoc* (if available) predictions for these records. The value on the x-axis is the time since the most recent dose (ADDL doses are included in the calculation of time-after-dose). If the "Display IDs" option in a graphic script is selected, a second page of graphics replaces points with subject IDs.

Version 4.4.2. Released 10 March 2012*New features:*

1. Standardized visual predictive check: This graphic is now produced automatically if a VPC is performed.
2. Prediction-corrected visual predictive check: This graphic is now produced automatically if a VPC is performed.
3. Alternate bootstrap approach: The traditional bootstrap approach constructs new datasets that have the identical number of subjects as the original dataset using
`sample(ALLIDS, n=ORIGINALN, replace=TRUE)`
An alternate approach allows the user to select smaller sample sizes:
`sample(ALLIDS, n=SMALLERN, replace=FALSE)`
This can be useful in evaluating the impact of different sample sizes on the results of an analysis.

Version 4.4.1. Released 29 January 2012*New features:*

1. Establish a minimum x-axis span: A minor modification to the Control Stream adjusts the span of the x-axis of By-Subject graphics to be no smaller than the requested values.
2. A user-configured file now allows the user to have > 20 Control Streams in the WORKINGFOLDER.

3. VPC Graphics: The user can now configure VPC graphics so that graphics in the log domain are suppressed.
4. VPC Graphics: VPC graphics can now be stratified by a within-subject marker (in addition to the already-existing stratification by a between-subject marker).

Version 4.4. Released 16 January 2012*New features:*

1. Adding data records to smooth graphics: Adding one line of code to the Control Stream allows **PLT Tools** to add records to the dataset to provide additional post-dose predictions, yielding smoother Cp profiles. Some constraints apply to the organization of the dataset (*e.g.*, CSV format; column names appearing in the first row of the dataset).

Version 4.3.1. Released 29 November 2011.*New features:*

1. LOQV: If the dataset contains a column LOQV and this column appears in the AllRecords file, LOQ values are obtained from this entry rather than from the corresponding field in the Graphics Script.
2. Global Header: If a Graphics script is not created or if the Global Header field in a Graphics Script is omitted, the \$PROBLEM statement is used to create a global header for graphics.

Version 4.2.0. Released 15 July 2011.*New features:*

1. Combining bootstrap runs: The user can now piece together the results from more than one bootstrap analysis.
2. Combining likelihood profiles: The user can now piece together the results from more than one likelihood profiles.
3. Restart from MSFI: If the user has invoked the "STOP NONMEM" button, the run can be restarted automatically using a model specification file.

Version 4.0.0. Released 15 May 2011.*New features:*

1. xpose. xpose graphics now contain headers and footers in the same style as graphics created by **PLT Tools**.
2. STOP NONMEM button: During NONMEM runs (NONMEM version 7.10 or greater), clicking this button initiates an orderly shutdown of NONMEM (by resetting MAXEVALS to a smaller value).

3. NONMEM 7.2: Complete support for NONMEM 7.2, including a feature to speed startup.
4. Project Log: Runs can now be selected using a Search tool.
5. Keyboard commands for Project Controller: Keyboard commands (shortcuts) permit the user to do the following:
 - a. Edit a Control Stream
 - b. Edit a Graphics Script
 - c. Run "NONMEM + Graphics"
 - d. Run "NONMEM Only"
 - e. Run "Graphics Only"

Version 3.5.0. Released 24 March 2011.

New features:

1. xpose. Complete support for xpose graphics added. This includes managing of NONMEM tables named with xpose naming conventions (e.g., sdtab), creation of xpose tables from AllRecords, management of aliases, creation of xpose graphics from pull-down menus, and support for user-written scripts containing xpose commands.
2. Graphics: histograms and qq plots for parameters and ETAS added to graphics.
3. A new "within-group" grouping variable added to Graphics Script.
4. NONMEM 7.2: Although NONMEM 7.2 is not yet released, complete support has been added based on final beta versions. This includes a new compiler option (pull-down menu item on Workspace/Options tab).

Version 3.4.0. Released 26 January 2011.

New features:

1. VLC (automated vertical lines in by-subject graphics) added.
2. Support for NONMEM 7.2 added (nmfe naming convention changed; new tables).

Version 3.3.0. Released 30 November 2010.

New features:

1. Networking capabilities greatly enhanced including simplified setup procedure.

Version 3.1.0. Released 15 November 2010.

New features:

1. CHECKOUT option in Control Stream: A Control Stream is now provided to facilitate use of NONMEM's CHECKOUT option.

2. FROMDATASET: A new option is permitted for the \$INPUT record in the Control Stream. If this option is invoked, **PLT Tools** replaces the text FROMDATASET with a list of data items obtained from the first row of the dataset.

Version 3.0.3. Released 26 October 2010.

New Features:

1. NUMSIGDIG: If NONMEM aborts before achieving 3 significant digits (or a different value, if selected as an option) and more than 0 digits, it reports the number of significant digits for each parameter estimated. These values are now incorporated into the Brief Summary document.
2. ETA shrinkage: For each ETA term displayed in graphics, ETA shrinkage is calculated. The calculation ignores values of ETA equal 0 (the approach recommended by Karlsson [personal communication]), in contrast to NONMEM's approach (which is to calculate ETA shrinkage using all values).

Version 3.0.0. Released 29 April 2010.

New Features:

1. Covariate Search: Complete support for an automated covariate search added.
2. NONMEM Help: NONMEM help files, provided as part of the NONMEM installation, can be accessed from within **PLT Tools**.
3. Batch Processing: Automation scripts can be generated from **PLT Tools**, permitting the user to repeat large numbers of runs in an automated manner. In addition, **PLT Tools** can use a different dataset if appropriate.
4. SAS Datasets: **PLT Tools** can reformat datasets to SAS XPORT, the format requested by FDA. This procedure can be applied to either or both all FDATA files created by NMTRAN or to all datasets archived by Archive Manager.
5. By-Subject Graphics: Concentration in a compartment with no predictions (*e.g.*, Cp or Ce when the only observations are an effect or concentration of a parent when all observations related to a metabolite) can be displayed.
6. By-Subject Graphics: Previously, when a subject had no observations, the subject was omitted from by-subject graphics. Now (except when a within-subject grouping variable is used), the subject is displayed with the text "No Observations". This enables the user to identify subjects with no observations.

Changes to Existing Features:

1. Run Listing: Improved user interface to access runs.

Version 2.5.0. Released 20 January 2010.

New features:

1. Support for NMQUAL. **PLT Tools** now allows the user to select either the Globomax / ICON installer (nmfe) or the Metrum installer (nmqual). Full support is provided for both. XML files created by nmqual are saved in a designated folder.
2. Covariate Search: Univariate covariate searches are now supported.
3. Single panel graphics: During assembly of routine graphics, in addition to creating a PDF document, **PLT Tools** creates single panel graphics (JPEG format) for a number of graphics that the user might consider embedding in a report.
4. Summary tables: At the completion of each NONMEM run, **PLT Tools** creates tables summarizing population and *post hoc* parameters.

Changes to Existing Features:

1. Report: The "Assemble Report" Tool (previous called "Prepare Report") includes many new features.
2. Graphics Editor: Several windows have been reorganized to improve clarity.

Version 2.4.0. Released 19 November 2009.*New features:*

1. Report preparation: The "Compare Runs" tab has been replace with "Prepare Report". In addition to the features existing previously, **PLT Tools** now generates a report in RTF (rich text) format. This report can be opened with either Microsoft Word or Open Office and contains graphics, text, and tables related to one or more NONMEM runs.
2. Report preparation, Archive Manager: Run # (timestamps) used for report preparation and archiving can be provided either from the Run Listing, a subset of the Run Listing (see next item), or using a pull-down menu to select a single run.

Changes to Existing Features:

1. Run Listing: A new column "Special" was added to the Run Listing editor. This column allows the user to identify a particular run as Base, Other, or Final.
2. Archive Manager: A new option permits the user to organizer the archive by File Type (the only choice available previously) or by Timestamp (run #).
3. Archive Manager: If analyses are performed on a non-Windows machine, the user can convert textfiles in the archive to Windows line endings.
4. Graphics Editor: Several windows have been reorganized to improve clarity.
5. Graphics Editor – LINK: If the user populates some or all of the LINK column in the ETAS area of a Graphics Script, labels for those entries are obtained from the THETAS area. This facilitates populating a Graphics Script.

Version 2.3.0. Released 14 September 2009.

New features:

1. NONMEM 7: NONMEM 7 is supported. The user selects NONMEM 5, 6, or 7 in Workspace / Options.
2. Progress Bar: A progress bar in the Project Controller window shows progress of loading of code.
3. First run: During the first run, **PLT Tools** assists the user in identifying the location of R, then selects a Working Folder containing a trial-run Control Stream.

Changes to Existing Features:

1. Startup: Startup of the Project Controller has been reorganized to speed startup markedly.

Version 2.2.0. Released 14 August 2009.

New features:

NMTRAN Errors: When NMTRAN reports error in the Control Stream, a pop-up window displays the error.

Version 2.1.2. Released 20 July 2009.

New features:

1. "Time after dose" graphics: If the text "Time after dose" appears in the header text for a spaghetti graphic, a new set of graphics is created in which observations are displayed relative to the most recent dose.
2. Seed for bootstrap and jackknife analyses: The user can enter a seed to control random-number generation for bootstrap and jackknife analyses.

Changes to Existing Features:

1. Automation: Certain environment variables are now allowed in file paths.
2. By-subject graphics in the presence of "reset events": Under the following set of events:
 - a. "reset" events (EVID = 2 or 3) were present in the dataset
 - b. the TIME data item was not monotonic increasing
 - c. the user did not provide information in the dataset and Graphics Editor regarding "multiple period"

Previously, **PLT Tools** displayed multiple lines in the same panel to accommodate these different dosing periods. Now, **PLT Tools** assigns sequential periods, documented in sub-header text.

Version 2.1.0. Released 25 May 2009.

New features:

1. Automation: Automation is now supported: the user creates a script (templates provided), then **PLT Tools** runs sequentially until complete.
2. Colored text in Output window: Text in the output window is now colored to indicate its source.
3. Improvements to Analysis window: Messages sent to the Analysis window were revised markedly.
4. THETA and OMEGA names in Brief Summaries: If the user enters (optional) THETA# values in the Graphics Editor, these names now appear in Brief Summaries. A checkbox allows the user to unselect this option in the compiled Brief Summaries created by "Compare Runs".
5. Display IDS (in graphics): Certain graphics display observations vs. predictions, the ratios of observations to predictions vs. time, or the differences between observations and predictions vs. time. If the graphics display outliers, the user may wish to use the graphics to identify those subjects whose values are outliers. This is accomplished by selecting the "Display IDs" option in the Graphics Editor (General Layout / Options tab). Additional graphics will be created in which IDs appear at all observations.

Changes to Existing Features:

1. Cleanup after "STOP". In previous versions in Windows, use of the "STOP" button could leave R running; in addition, the file "NONMEMisRUNNING.txt" might not have been deleted. Revisions to the code now allow proper close-out of a run after use of the "STOP" button.

Version 2.0.4. Released 6 March 2009.

New features:

1. Installer: Installers are now provided for Windows and OS X.
2. Real-time parameter displays: When R is used, a window displays real-time updates of the objective function and THETAS.

Version 2.0. Released 15 December 2008.

New features:

1. Data checkout graphics: Graphics are prepared showing observations (by subject, composite [and by dose group, also, linear and log scales]), and dose-normalized (linear and log scales).
2. Jackknife: **PLT Tools** can perform a jackknife analysis.
3. Project Log: After each NONMEM run is completed, a Project Log is updated with the timestamp, name of the Control Stream, \$PROBLEM statement, the objective function, the

number of significant digits and other characteristics of the run. Clicking on any line in the window enables buttons at the bottom of the window; these buttons open the Graphics, Brief Summary, Summary or Results files for that run.

4. Parameter Log: After each NONMEM run is completed, a Parameter Log is updated with the timestamp and the values of all THETAs, OMEGAs, and SIGMAs.

Changes To Existing Features:

1. Reorganization of the workspace to facilitate throughput.

Version 1.7. Released 21 October 2008.

New features:

1. Control Stream Editor: A new button in the Project Controller permits the user to edit a Control Stream directly from **PLT Tools**.
2. Control Stream Library: A new button in the Project Controller permits the user to access a library of Control Streams, then save the template in the Working Folder. The file can then be edited using the Control Stream Editor (or any editor preferred by the user).
3. Likelihood Profile: **PLT Tools** can perform a likelihood profile.

Version 1.6. Released 14 October 2008.

New features:

1. View Run Log: This facilitates assembly of the Run Listing for Compare Runs and Create Archive.
2. Workspace / Options: Selection of the Working Folder (and Project Folder) was moved to a new tab. All tabs in a single window are now linked to the same Project Folder / Working Folder. Location of options is reorganized to simplify their selection.
3. Graphics Editor: A new tab, Text Mapping, has been added to simplify text mapping.
4. Clear console on each run: A new option clears the Output window on each run.

Bug Fixes:

1. Error messages: Error messages are now improved. In particular, most error messages now appear in pop-up message boxes.

Version 1.5. Released 24 September 2008.

New features:

1. Print Objective Function: Selecting this option results in the minimum value of the objective function being printed on each page of graphics.

2. Print Control Stream: Selecting this option results in the name of the Control Stream being printed on each page.
3. Wider margins: Selecting this option results in graphics having a wider margin (1" on all sides, compared to ½" without the option).
4. NONMEM Only: Clicking this button causes NONMEM to be executed with no attempt to create graphics.
5. Analysis Conducted In: This field displays the folder in which NONMEM is being executed. It is relevant only for registered users who are configured for simultaneous conduct of NONMEM runs or for network access.

Changes To Existing Features:

1. Echoing to Output window: Procedures for echoing to the output window are improved.
2. Error messages: Serious errors now appear in a pop-up box to ensure that the user is aware of the message.
3. Location of PLTTools folder in Windows: The PLTTools folder can now be located on any drive from C – Z (previously limited to C – K).

Bug Fixes:

1. NMSEE: The Windows version of nmsee.exe appeared to be corrupted under certain circumstances. A new version is provided and the user is encouraged to replace the existing version during installation.

Version 1.4. Release 1 August 2008.

New features:

1. Bootstrap: Fully automated support for bootstrap.
2. Visual predictive check: Full automated support.
3. Consultant mode: Allows registered users to revise the text that appears in the
4. User-defined scripts: The user can write his / her own scripts in R. These scripts can access "objects" such as ALL (the contents of AllRecords), ETAS (the contents of FirstRecords), DEMO (the contents of the covariates file), GlobalHeader (the user-selected header entered into the Graphics Editor) to create additional graphics or perform statistical analyses.

Changes To Existing Features:

1. Cleanup procedures: If a job is terminated prematurely, cleanup of remnants is improved.

Version 1.3. Released 15 June 2008

New Features:

1. NONMEM 5: **PLT Tools** is now compatible with NONMEM 5. See the section on NONMEM 5 earlier in this manual for implementation.
2. NONMEM 6 version 2.0: **PLT Tools** is compatible with the most recent release of NONMEM.
3. Fortran compatibility: Additional Fortran compilers have been evaluated successfully with **PLT Tools**. See Installation Manual for further information.
4. Use of R: Previous versions of **PLT Tools** required that registered users use S-Plus rather than R. The Free Software Foundation (home of the GNU Project) has determined that **PLT Tools** does not violate R's GPL (General Public License). As a result, all users of **PLT Tools** can now select freely between R and S-Plus (support for S-Plus was discontinued at a later version).
5. "Local Only" option: This feature applies only to users in a networked environment. If **PLT Tools** is normally run on a network and the computer on which it is run is removed from the network (e.g., a laptop computer is taken home), **PLT Tools** will no longer attempt to access the network.
6. Cleanup Script: Clicking the "Stop" button in **PLT Tools**'s Project Controller window in earlier versions left certain file remnants behind. Now, a "cleanup script" is launched to delete these files. In networked versions, if certain remnants cannot be deleted, informative messages are sent to the Project Controller window.

Changes to Existing Features:

1. NONMEM output: In certain situations, NONMEM output was not sent to the Output window in the Project Controller. This is corrected automatically with installation of NONMEM VI 2.0. For other versions of NONMEM, the nmfe batch file must be edited; instructions for doing so are sent to the Output window.
2. Network processing: Procedures for conducting analyses using multiple CPU's on a network have been improved markedly.
3. Graphics Script: The Project Controller window includes a button permitting the user to create a new graphics script without accessing the pull-down menus.

Bug Fixes:

1. Graphics: Certain error conditions are now trapped and the user is notified.

Version 1.2. Released 1 May 2008

New Features:

1. Conditional weighted residuals are now supported. Certain modifications to the NONMEM code are required (see this manual) and a file "compute.cwres.R-PLTTools-V1", provided with this version of **PLT Tools** must be placed in the PLTTools folder. If conditional weighted residuals are calculated, they are displayed in graphics.

2. Parameter table: A new table, TABLES/PARAMETERS/Parameters.TIMESTAMP.csv (or .TAB) contains the parameters for that NONMEM run.
3. Simulation and simulation/estimation: These processes are now fully supported. In particular, if NSUBPROBLEMS > 1, separate graphics are prepared for each SUBPROBLEM. In addition, the composite tables created by NONMEM are separated into separate tables for each subproblem (and placed in a separate folder). Finally, parameters for each of the subproblems are aggregated in the table identified above.
4. Batch processing: If the user initiates a second run while NONMEM is already running, the second run is delayed until the first run finishes. This permits the user to stack several runs, e.g., at the completion of a work day.
5. Simultaneous processing: The user can perform more than a single NONMEM run simultaneously on a single computer. If the computer has multiple processors, this may permit more rapid throughput. Requires registration.
6. Network processing: The user can access a network of computers to perform multiple simultaneous NONMEM runs. Requires registration and a supplemental fee (contact Sales@PLTsoft.com for further information).
7. "Echo NONMEM": Certain Fortran compilers do not send their output to the "standard output" as the data are being recorded in the NONMEM outputfile. As a result, the user is unable to gauge the present status of a NONMEM run, particular if that run is lengthy and/or prone to aborting. A new feature in **PLT Tools** examines the NONMEM outputfile at regular intervals, then prints any new lines in the Output window. If the Fortran compiler sends its output to the Output window and this feature is selected, redundant text will appear; this can be fixed by de-selecting the option

Changes to Existing Features:

1. The Brief Summary document has been reformatted. In addition, the code used to construct the document has been revised significantly to accommodate a variety of scenarios including the use of \$PRED rather than \$PK and single-subject data.
2. A new set of graphics automatically displays conditional weighted residuals, provided that the user has modified the NONMEM code as described in this manual.
3. A new Graphics Script can be opened directly from the Project Controller window (the **New** button).

Version 1.1. Released 26 March 2008

New Features:

1. Default Graphics. If the user does not populate a graphics script (using the Graphics Editor), a set of default graphics is attempted, depending on the entries provided in the AllRecords and FirstRecords tables.
2. Naming of Tables: AllRecords.txt and FirstRecords.txt now allowed.

3. Requirements for Tables in Order to Create Graphics: If certain elements are omitted from tables, limited graphics can now be created.
4. Single-subject data: Brief Summary and Graphics now created correctly for single-subject data.
5. MDV, AMT: If MDV or AMT not provided in AllRecords but present in FDATA, MDV and/or AMT are copied to AllRecords.

Changes to Existing Features:

1. Multiple small changes involving spacing, formatting, text positioning.

Error Detection:

1. Incorrect IDs in Tables: If number of unique IDs in table does not correspond to number of individuals detected by NONMEM, error reported. This could occur if an ID contained > 6 digits and was truncated during TABLE step.
2. Failure to Select FIRSTONLY option: If the number of rows in FirstRecords is larger than the number of individual subjects detected by NONMEM (e.g., if the user omitted the FIRSTONLY only option in \$TABLE), a warning message is sent and certain graphics are omitted.
3. NOHEADER Option for Tables: Selecting the NOHEADER option in \$TABLE prevents **PLT Tools** from generating graphics. A warning is provided.

Version 1.0. Released 1 March 2008

APPENDIX 1. Setting Up Parallelization of NONMEM Runs

PLT Tools supports parallel runs in NONMEM using the `—parafile` option. At present, all parallel runs must be executed on the same computer. If a user wishes to execute parallel runs across multiple computers, please contact support@PLTsoft.com.

Executing parallel runs in NONMEM requires that the user install MPI software provided with all recent NONMEM installation disks. It is the user's responsibility to install and configure this MPI software. Once that software has been installed and the user has demonstrated that NONMEM can perform parallel runs, **PLT Tools** can be configured as follows:

1. In **Workspace / Options**, select the number of nodes that the user wishes to use. This is a non-stored option and will need to be selected each time **PLT Tools** is opened.
2. Place a single PNM file in any of the following locations: **PLTTools-Support**, **PROJECTFOLDER**, **WORKINGFOLDER**. **PLT Tools** searches these folders in that order, selecting the PNM file from the last location in which a PNM file is located.
3. If the user selects nodes > 1 and **PLT Tools** cannot identify a PNM file or identifies more than one PNM file, **PLT Tools** terminates immediately with an informative error message.
4. If the user selects nodes > 1 and **PLT Tools** identifies one PNM file, **PLT Tools** adds the appropriate options to the command that is issued to the operating system to execute NONMEM.
5. Users who do not have a license for **PLT Tools** are limited to 2 nodes at present. If a user wants to evaluate the capabilities of **PLT Tools** to support parallel runs with > 2 nodes, contact support@PLTsoft.com to obtain a temporary license. For licensed users, the limit on nodes is determined by the computer configuration, not by **PLT Tools**.

Aravind Mittur, Ph.D (Impax Specialty Pharma, Hayward, California) provided the following detailed explanation of his experience setting up parallelization of NONMEM runs and configuring **PLT Tools** for that purpose in Windows 7, Windows 8, and Windows 10.

1. MPICH2 packaged in the NONMEM distribution was set up during routine installation of NONMEM 7.3.0 and 7.4.1 (gfortran 4.6.3; .NET Framework 4.7.1). Each session of `mpiexec.exe` will require a username and password (typically the Windows or local server logon credentials). This can be avoided by encrypting the access information into Windows registry as follows:
 - > `c:\nm74g64\run>mpiexec -register`
 - > account (domain\user):
 - > password:
 - > confirm password:
 - > Password encrypted into the Registry.
2. Add MPICH2 path to Windows system environment variables:
 - a) Right click "My Computer" and pick properties
 - b) Select the Advanced Tab or Advanced System Settings
 - c) Select the Environment Variables button

- d) Edit the path variable under System Variables to add "C:\ Program Files\MPICH2\bin" (or pertinent location) before NONMEM and gfortran (or the pertinent compiler), **separated** from the prior and subsequent path with a semicolon.

3. Test MPICH2 installation using example cpi.exe included in the distribution

```
> c:\nm74g64\run>mpiexec -n 4 cpi.exe
```

```
Enter the number of intervals: (0 quits) 4
```

```
pi is approximately 3.1468005183939427, Error is 0.0052078648041496
```

```
wall clock time = 0.000117
```

```
Enter the number of intervals: (0 quits) 99999999999999999999
```

```
pi is approximately 3.1415926535898331, Error is 0.00000000000000400
```

```
wall clock time = 2.725859
```

Results confirm execution and installation of MPICH2. If MPICH2 fails to launch, check and ensure PATH in environmental variables was inclusive, also test by executing command with the absolute path to MPICH2.

4. Next, test NONMEM examples within the command shell using MPI for a single computer with multiple cores as suggested in section I.61 of the NONMEM USERS GUIDE - INTRODUCTION TO NONMEM 7.4.1 (Robert J. Bauer, ICON Plc, May 23, 2017) (control streams and datasets are located in the NONMEM install directory, example C:\nm74g64\run. The example parafile "mpiwini8.pnm" was edited to reflect the number of cores available at hand.

```
> c:\nm74g64\run>Nmfe74 foce_parallel.ctl foce_parallel.res -parafile=mpiwini8.pnm  
[nodes]=4
```

Ran fine, no issues, confirmed output matches that in the distribution.

5. Next, tested running the same control stream in parallel mode but edited to print AllRecords.txt and FirstRecords.txt:

```
> c:\nm74g64\run>Nmfe74 foce_parallel_AM.ctl foce_parallel.res -parafile=mpiwini8.pnm  
[nodes]=4
```

Confirmed that NONMEM detects request for parallel mode operation, verified utilization of all 4 cores in the Performance Monitor, no run issues, AllRecords.txt and FirstRecords.txt were created.

6. Next, tested the same parallelized example foce_parallel_AM.ctl using PLTTools (v5.5.1, Build Date: 2018-02-13 07:24:18, R x64 v3.4.2) as follows:

- a) Moved foce_parallel_AM.ctl and example1b.csv to a test folder
- b) Copied "mpiwini8.pnm" to "testpnm.pnm" in PROJECT\WORKING. Edit the PNM file to customize parallelization options. As noted above, PLTTools requires a single PNM file called "testpnm.pnm" to be present in the WORKING folder (multiple PNM files can reside in subfolders)
- c) Edited control stream to point to the correct data folder, added \$TABLES record
- d) Choose NODES = 4 (number of cores available) in Workspace/Options. Note that the number of cores chosen in PLTTools overrides the number of nodes (cores) selected in

the PNM file. For compute intensive runs, it is suggested that only N-1 cores be utilized to facilitate smooth and continual operation of other Windows applications including multiple instances of PLTTools.

Run completed, all pertinent output including Tables and Graphics were created, and match the standard outputs. Confirms installation and parallel processing of NONMEM runs using PLTTools.

Control File from NONMEM/RUN folder used for testing MPI

```

; Used for comparing single versus parallel computing for FOCE method.
;$SIZES LVR=30
;$SIZES LTH=15
;$SIZES LIM1=100
$PROB RUN# Example 1 (from samp51)
$INPUT C SET ID JID TIME DV=CONC AMT=DOSE RATE EVID MDV CMT CLX V1X QX
V2X SDIX SDSX
$DATA ../DATAFILES/example1b.csv IGNORE=C
$SUBROUTINES ADVAN3 TRANS4
$PK MU_1=THETA(1)
      MU_2=THETA(2)
      MU_3=THETA(3)
      MU_4=THETA(4)
CL=DEXP(MU_1+ETA(1))
V1=DEXP(MU_2+ETA(2))
Q=DEXP(MU_3+ETA(3))
V2=DEXP(MU_4+ETA(4))
S1=V1
$ERROR
Y = F + F*EPS(1)
IPRED = F
IRES = DV-IPRED
$THETA (0.001, 2.0) ;[LN(CL)]
      (0.001, 2.0) ;[LN(V1)]
      (0.001, 2.0) ;[LN(Q)]
      (0.001, 2.0) ;[LN(V2)]
$OMEGA BLOCK(4)
0.15 ;[P]
0.01 ;[F]
0.15 ;[P]
0.01 ;[F]
0.01 ;[F]
0.15 ;[P]
0.01 ;[F]
0.01 ;[F]
0.01 ;[F]
0.15 ;[P]
;Initial value of SIGMA
$SIGMA (0.6) ;[P]
;$EST METHOD=CHAIN FILE=foce_parallel.chn ISAMPLE=3 NSAMPLE=0
$EST METHOD=1 INTERACTION NSIG=3 PRINT=1 NOABORT MAXEVAL=9999
FORMAT=,1PE23.16
MSFO=foce_parallel.msfo ;PARAFILE=parallel_file.pnm ATOL=5
$COV MATRIX=S PRINT=E UNCONDITIONAL ; PARAFILE=OFF
$TABLE NOPRINT ONEHEADER FILE=ALLRECORDS.TXT
      ID TIME DV DOSE RATE EVID MDV CMT PRED IPRED
$TABLE NOPRINT FIRSTONLY NOAPPEND FILE=FIRSTRECORDS.TXT
      ID CL V1 Q V2 ETA1 ETA2 ETA3 ETA4

```


PNM File from NONMEM/RUN folder edited and used for testing MPI (testpnm.pnm, Mittur changes highlighted)

```
$DEFAULTS
; User may specify their own variables with bracketed words at the
nmfe72 script command line:
; nmfe72 myprog.ctl myres.res "-parafile=mpiwini8.pnm" "[nodes]=3"
; which will over-ride default settings of variables listed here
(variables must be defined one
; variable per line). If the file defaults.pnm exists, and it defines
[nodes], this can also
; over-ride defaults listed in the parafile.
; Order of over-ride is Command line on nmfe72 script over-rides
defaults.pnm,
; which over-rides defaults defined in parafile.
; The advantage to this ordering is that, a generic parafile file can
be created for most environments.
; A user may then over-ride defaults specified in this generic
parafile with his own in defaults.pnm,
; that may be more suitable to his environment. Finally, a user can
temporarily over-ride his own defaults
; by giving an alternative value as an nmfe72 script command option.
[nodes]=4
$GENERAL
; [nodes] is a User defined variable
; COMPUTERS=2
NODES=[nodes] PARSE_TYPE=2 PARSE_NUM=200 TIMEOUTI=600 TIMEOUT=1000
PARAPRINT=0 TRANSFER_TYPE=1
; SINGLE node: NODES=1
; MULTI node: NODES>1
; WORKER node: NODES=0
; parse_num=number of subjects to give to each node
; parse_type=0, give each node parse_num subjects
; parse_type=1, evenly distribute numbers of subjects among available
nodes
; parse_type=2, load balance among nodes
; parse_type=3, assign subjects to nodes based on idranges
; parse_type=4, load balance among nodes, taking into account loading
time. Will assess ideal number of nodes.
; If loading time too costly, will eventually revert to single CPU
mode.
; timeouti=seconds to wait for node to start. if not started in time,
deassign node, and give its load to next worker, until next iteration
; timeout=minutes to wait for node to complete. if not completed by
then, deassign node, and have manager complete it.
; paraprint=1 print to console the parallel computing process. Can
be modified at runt-time with ctrl-B toggle.
; But parallel.log always records parallelization progress.
; transfer_type=0 for file transfer, 1 for mpi
; THE EXCLUDE/INCLUDE may be used to selectively use certain nodes, out
of a large list.
; $EXCLUDE 5-7 ; exclude nodes 5-7
```

```
;$EXCLUDE ALL
;$INCLUDE 1,4-6
$COMMANDS ;each node gets a command line, used to launch the node
session
; %* sends all arguments on the user's command line to the manager
process
1:mpiexec -wdir "%cd%" -localonly -n 1 nonmem.exe %*
; Only specific arguments should be sent to the workers, which are
identified by reserved variable names
2-[nodes]:-wdir "%cd%\worker{#-1}" -n 1 nonmem.exe
$DIRECTORIES
1:NONE ; FIRST DIRECTORY IS THE COMMON DIRECTORY
2-[nodes]:worker{#-1} ; NEXT SET ARE THE WORKER directories
$IDRANGES ; USED IF PARSE_TYPE=3
1:61,100
2:1,60
```

Disclaimers

PLT Tools has undergone extensive testing and is designed to allow the user great flexibility in the data from which graphics are created. However, because of the remarkable flexibility of NONMEM and the lack of standardization of NONMEM's output tables, PLTsoft cannot be responsible for errors in graphics. However, if the user follows the recommendations in this manual, extensive testing indicates that the results will be correct.