Learn About Pirana

An Overview of Pirana	**General Information**
Main window Projects Folders Folder contents Files Models	Release Notes for Pirana 3.0/RsNLME 1.0.0 (2020) Known Issues Support website Training website PDF List
Data Inspection	**Analyzing Results and Output**
Data Inspector Correlation calculator Check dataset	View and compare parameter estimates View execution log Calculate AIC/BIC Matrices R scripts for graphs and file processing
NLME Model Management	**NONMEM Model Management**
Create or edit a model using Model Builder Generate a new model using a template Run initial estimates Duplicate models Batch editing of models Open the input dataset Attach folders to selected model Rename a model file View difference with reference model Delete a model Fit a model Execute a model run with Model Executor View model results	Create NONMEM models Add notes and view model properties Duplicate models Batch editing of models Create dataset template R script Open the input dataset Attach folders to selected model Rename a model file Generate NM parallelization file Generate SCM configuration file Execute a model View intermediate results of active runs Model translation Import results from a model directory Export model items Create zip file of model items Diagnostic graphs with Xpose NONMEM VPC plots with PsN and Xpose Automated Modeling Workflow Example
Reporting Functionality	**Installation and Configuration**
Run reports Run records Visual Run Record	Pirana Installation Pirana and Clusters Configuration Settings Keyboard shortcuts Command line parameters Troubleshooting/FAQ

Content

An Overview of Pirana

Pirana is a modeling workbench for RsNLME, NONMEM, and PsN, offering a graphical user interface and many auxiliary tools to support modeling and simulation analyses. It can be used for modeling on a local system or on computer clusters. Pirana can be used to run, manage, and edit models, interpret output, and manage NONMEM installations. It is easily extendable with custom scripts, and integrates smoothly with R, Xpose4, Excel and other software.

Model management is enhanced through descriptions, notes, and coloring of models and results lists. Pirana provides condensed and detailed model information displays, as well as both list and tree views. Pirana offers wizards and templates to aid in model creation, duplicates models with updated run- and table numbers and parameter estimates, and will delete model files and all associated results and table files.

In terms of **results management**, Pirana creates HTML and LaTeX run reports that are formatted and contain basic model specifications and estimations results for all estimation methods that were used, including parameter estimates, uncertainty, shrinkage etc. Custom R-scripts can be used from within Pirana and run on a specific model, e.g., to automate creation of goodness-of-fit plots and automatically load the output. Pirana comes with multiple useful scripts, ready for customization. Pirana's Data Inspector allows detailed investigation of, e.g., goodness-of-fit plots, or plots of covariates against individual parameter estimates. Pirana's main window provides an overview of dataset files, output files, Xpose files, and R scripts. The data files and Xpose datasets can be opened and edited with a spreadsheet, code editor, or in R. Notes can be added to datasets. Pirana also has tools for converting between NONMEM table files and CSV format and converting $DES code to Berkeley Madonna or R-deSolve code for simulation purposes.

Pirana offers tools for local and cluster **NONMEM installation management**. NONMEM run progression can be monitored. Pirana also reads intermediate NONMEM output and provides numerical and graphical view of parameters and gradients

Execution of models using PsN is available through Pirana's PsN dialog, which accepts all PsN commands and allows selection of the NONMEM version to use. The actual command line that is used is displayed and can be edited. The dialog also shows all PsN information for the specific command. Pirana also contains a wizard for creating an scm configuration file.

Pirana **supports clusters** and has wizards for the creation of configuration files for NONMEM parallel computation features. Computer clusters running NONMEM can be accessed directly through SSH, both from/to Linux and Windows systems. Pirana can be installed on the cluster server and run by multiple clients through SSH-X-window tunneling.

Additional information is available on the following topics:

- Main window
- Projects
- Folders
- Folder contents
- Files
- Models

Main window

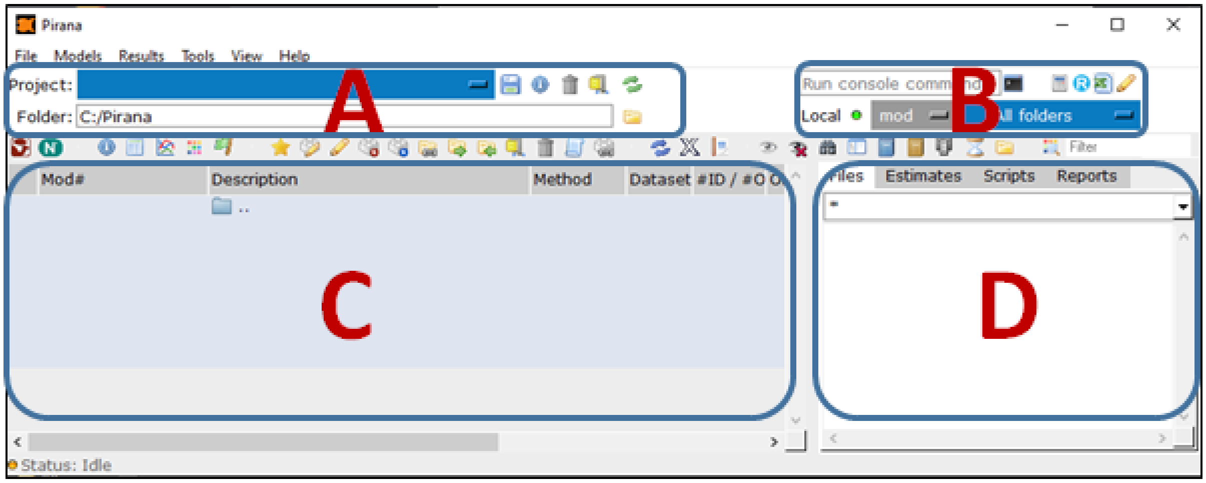
The *Pirana* window has four main areas, in addition to the Menus and Toolbar:

Area A contains project navigation and management tools.

Area B has tools for opening different applications, e.g., a console window ( ), calculator ( ), spreadsheet application ( ), or text editor ( ). There also tools for controlling the models listed in area C. (Note that entering a command in the field next to and pressing the <Enter> key on your keyboard executes the command and opens the console window.)

Area C is a table of the folders (listed first in the table) and models in the current project. Referred to as the “model overview”.

Area D contains tabs that list files (see “Files tab”), parameter estimates (see “View and compare parameter estimates”), R-scripts (see “R scripts for graphs and file processing”), reports (Reports tab).



There are tooltips for many of the icons, buttons, and other parts of the user interface. Hover the mouse cursor over the item to display any tooltips.

The visibility of the toolbar is controlled using the **View > Show toolbar** menu option or through the right-click menu of a model (**Batch operations > Show toolbar**).

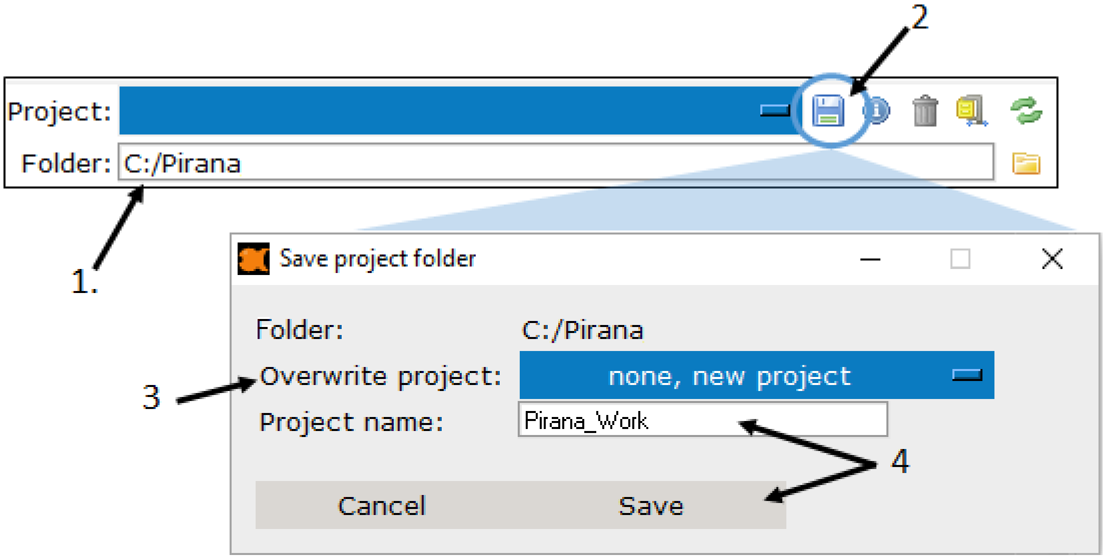
Projects

A project, in Pirana, is a link to a folder that enables you to quickly switch between folder contents.

- Create new project
- Switch projects
- View/Edit project properties
- Save project to a zip file
- Reload models in project
- Delete project

Create new project

1. Identify the folder the link as a project by typing the full path in the **Folder** field.
   Or
   Click 📁 to browse to the folder.
   Or
   Double-click the folders in the model overview area to navigate to the folder.
2. Click 💾.
3. In the dialog, select the action to take if the project name already exists from the **Overwrite project** pull-down.
4. Enter a **Project name** in the field.
5. Press **Save**.


Project:
Folder: C:/Pirana

Save project folder

Folder: C:/Pirana
Overwrite project: none, new project
Project name: Pirana_Work

The new project is now available for selection in the **Project** pull-down.

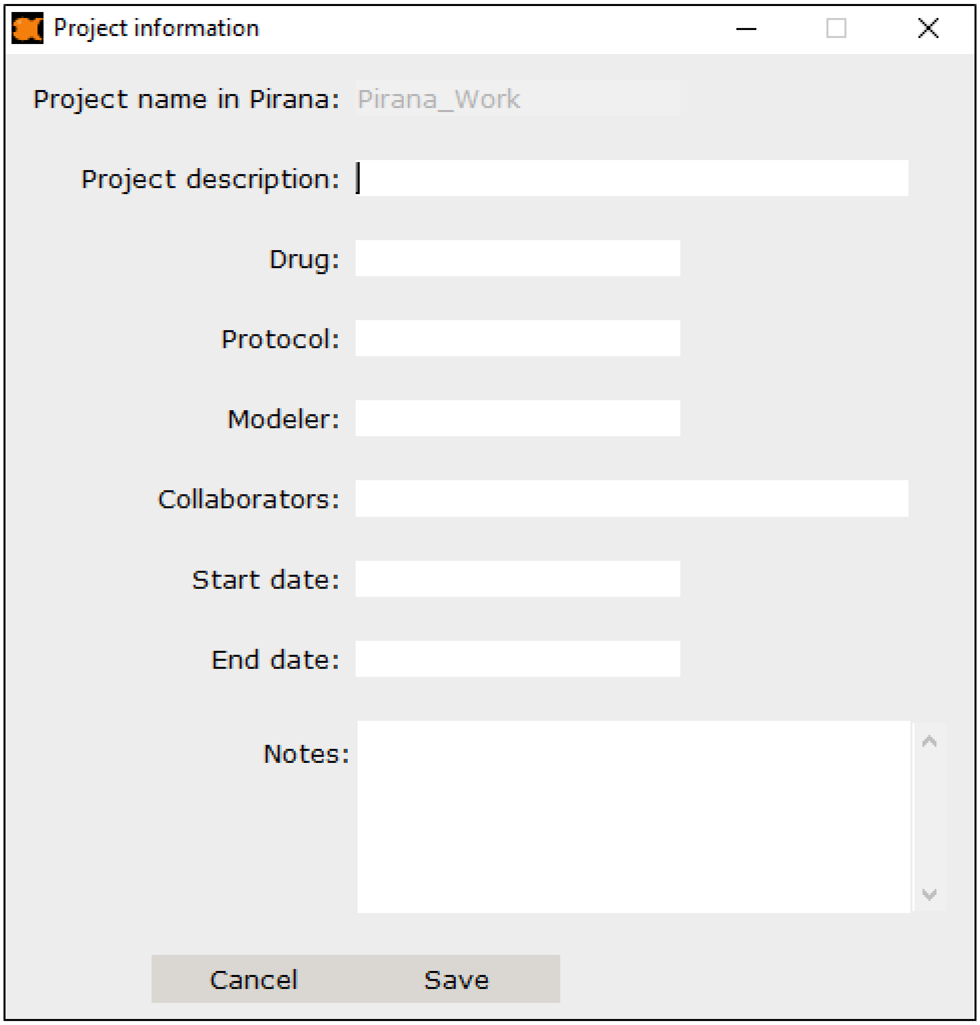
Switch projects

Select a different project using the **Project** pull-down.

All folders, models, and files in the selected project are loaded into the *Pirana* window.

View/Edit project properties

1. Click ⓘ next to the **Project** pull-down to view and change any project details.



Project information

Project name in Pirana: Pirana_Work

Project description:

Drug:

Protocol:

Modeler:

Collaborators:

Start date:

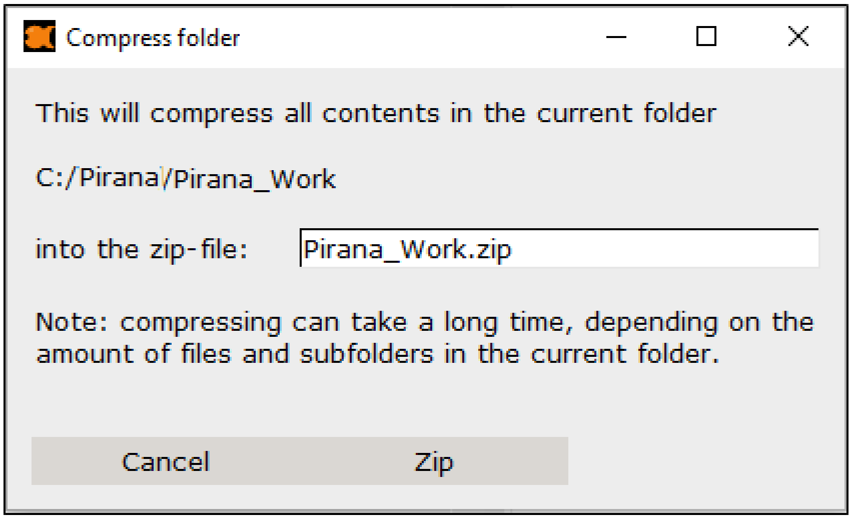
End date:

Notes:

2. Press **Save** to accept any changes.

Save project to a zip file

1. Click next to the **Project** pull-down to compress the entire project and save it as a zip file.


Compress folder

This will compress all contents in the current folder

C:/Pirana/Pirana_Work

into the zip-file: Pirana_Work.zip

Note: compressing can take a long time, depending on the amount of files and subfolders in the current folder.

2. In the dialog, enter a name for the zip file.
3. Press **Zip**.

Reload models in project

Click next to the **Project** pull-down to reload contents of the selected project.

A refresh/reload should be done when you make changes to models or add files outside of Pirana. Also when a run is finished, you should refresh to gather the results into Pirana.

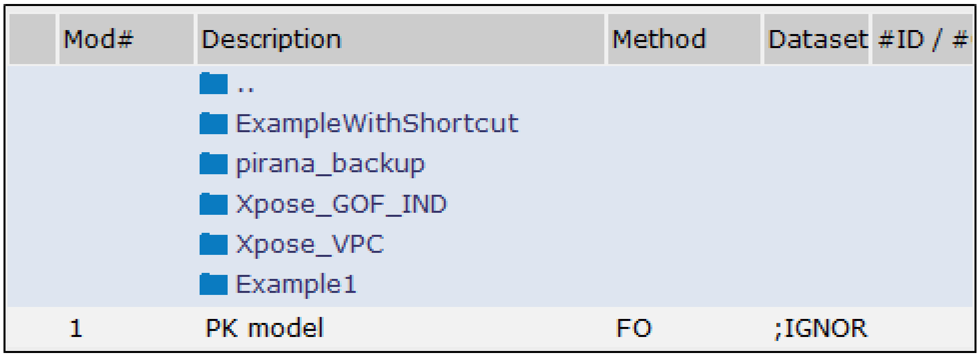
Delete project

To delete a project from the list, click next to the **Project** pull-down.

The link to the folder is removed, but the folder and its contents remain unchanged.

Folders

Folders can be used to help with organization of your files in a project. They are listed in the top section of the model overview area.

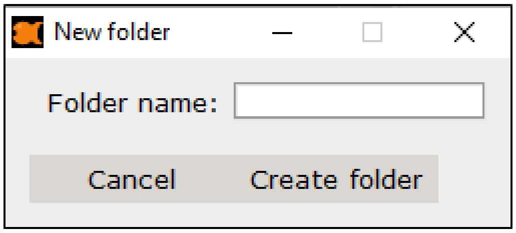


	Mod#	Description	Method	Dataset	#ID / #
		..			
		ExampleWithShortcut			
		pirana_backup			
		Xpose_GOF_IND			
		Xpose_VPC			
		Example1			
	1	PK model	FO	;IGNOR	

- Add folder
- Open folder in file explorer
- Rename folder
- Clean up folder
- Delete folder

Add folder

1. Select the **File > New Folder** menu option.



New folder

Folder name:

Cancel　Create folder

2. In the dialog, enter a name for the new folder.
3. Click **Create folder**.

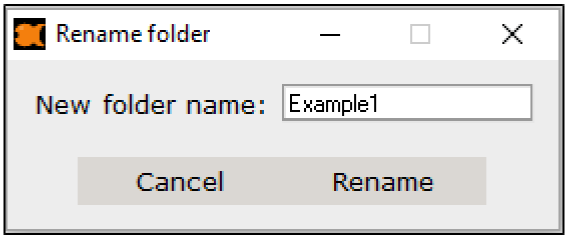
Open folder in file explorer

1. Select the folder in the model overview list.
2. Click 📁 in the toolbar.
   Or
   Select the **Tools > Open folder in file explorer** menu option.

The file explorer window opens displaying the folder contents.

Rename folder

1. Right-click the folder and select **File actions > Rename**.



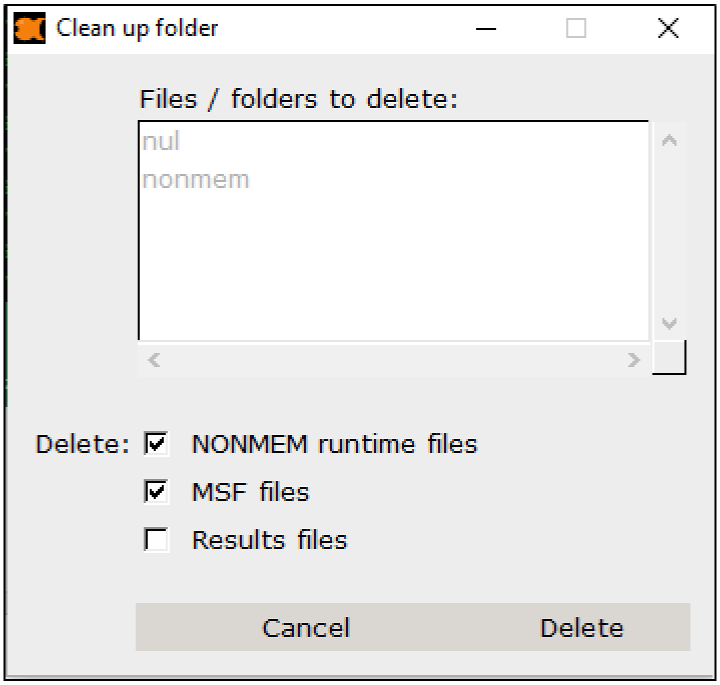
New folder name: Example1

2. In the dialog, enter the new name.
3. Press **Rename**.

Clean up folder

NONMEM runtime files, MSF files, and results files that are not used or no longer needed can be quickly removed.

1. Select the **Tools > Remove runtime files current folder** menu option.



Files / folders to delete:

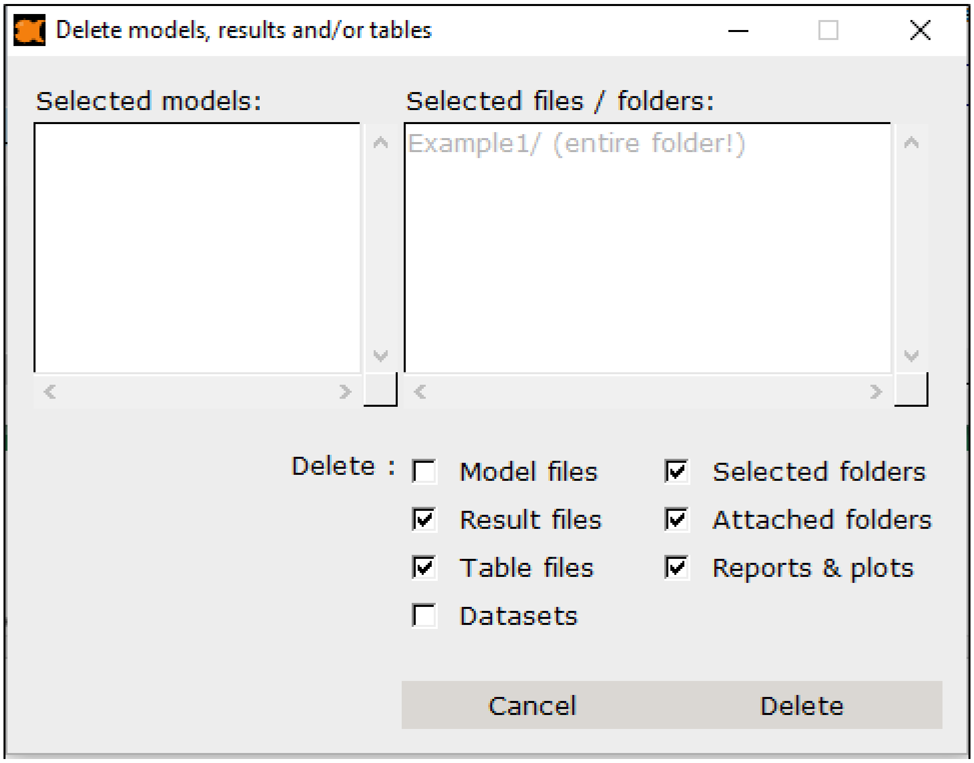
nul
nonmem

Delete: ☑ NONMEM runtime files
☑ MSF files
□ Results files

2. In the folder, check the boxes for the types of runtime files to delete. Any file or folder listed in the top section will be deleted.
3. Press **Delete**.

Delete folder

1. Select the folder to delete in the model overview list.
2. Click the 🗑 icon in the toolbar
   Or
   Right-click the folder and select **File actions > Delete**.



3. In the dialog, make sure the **Selected folders** box is checked.
4. Press **Delete**.

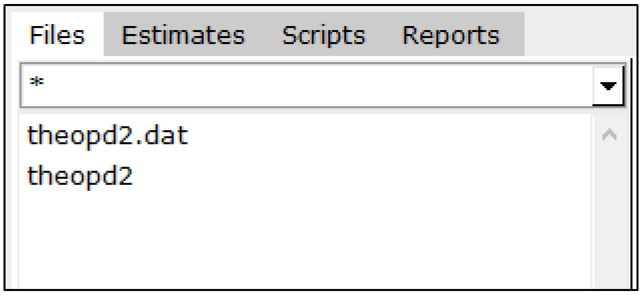
Folder contents

The right side of the window is where the file contents of folder being viewed on the left are listed under several tabs.

Note: It is best practice to click the **Refresh** icon (🔄) in area A (refer to the image in the "Main window" section) each time a new folder is selected for viewing.

Files tab

All files in the current folder are listed. Type in the field at the top to filter the list.



theopd2.dat
theopd2

Estimates tab

The Estimates tab lists the parameter estimates for the executed model selected in the model overview list. See "View and compare parameter estimates" for more information.

Scripts tab

The Scripts tab lists the R scripts that are available either in the **Project**, in the **Library**, or in **My scripts**. Use the pull-down in the tab's toolbar to select which set of scripts to list. For more information on this tab, see "R scripts for graphs and file processing".

Reports tab

The Reports tab lists the reports that have been generated for different runs in the project. See "Run reports" for more information about this tab.

Files

- Open files
- Edit file
- View/Edit file properties
- Convert between comma- and tab-separated format
- Copy and move files
- Rename files
- Delete files

Open files

To open a file in a particular application, in the Files tab on the right side of the window:

1. Select the file in the list.
2. Right-click the file and choose **Open in**, then select the application from the list in the submenu.

Choices include the Data Inspector, spreadsheet, the R GUI, the external editor, and the internal editor. The application associated with most of these options is determined using the Software integration panel of the *Settings* dialog.

Edit file

There are two options for file editors: built-in and external. The external text editor is defined in the Software integration Configuration Settings page. Use the **View > Use built-in model editor** menu option to toggle using the built-in model editor or the external one.

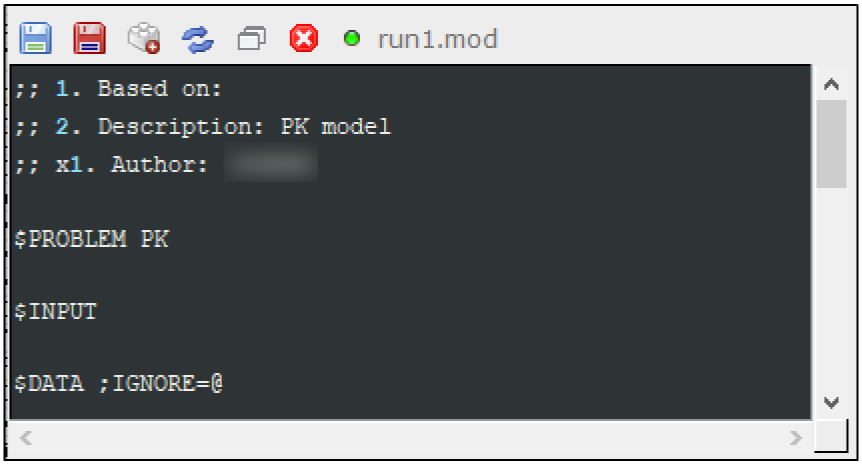
To edit a model file

1. Select the model in the list.
2. Double-click the model.
   Or
   Right-click the selected model and choose **File actions > Edit** from the menu.
   Or
   Select the **Models > Edit Model** menu option.
   Or
   Click in the toolbar.

Or, to edit a file that does not contain a model

1. In the Files tab on the right side of the window, select the file to edit.
2. Right-click the file and choose **Open in > External editor** or **Open in > Built-in editor** from the menu.
   Or
   Double-click the file to open the file in an external editor (e.g., .R files open in RStudio, if configured in the Software Integrations Settings dialog.

If **Built-in editor** is selected, the editor opens below the model overview area.



```
;; 1. Based on:
;; 2. Description: PK model
;; x1. Author: [redacted]

$PROBLEM PK

$INPUT

$DATA ;IGNORE=@
```

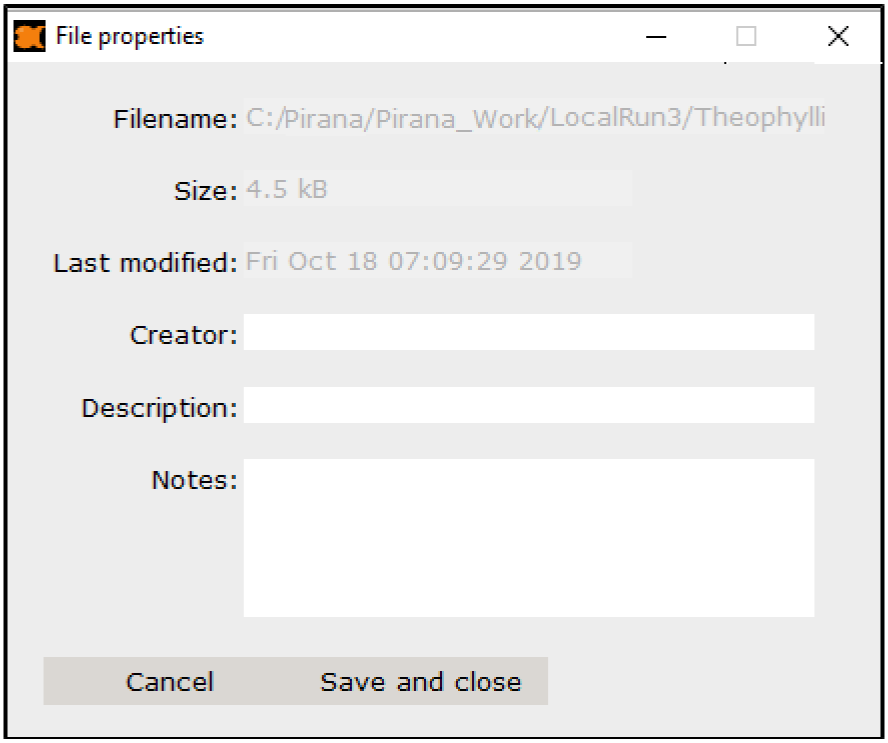
3. Type directly in the editor area to modify the file contents.
4. Use the icons in the editor toolbar to:

- Save the file edits (💾).
- Save the edits to a different file (💾).
- Create a duplicate file (⧉).
- Reload the file contents in the editor (🔄).
- Open the editor in a separate window (🗗).
- Close the editor (☒).

View/Edit file properties

To view the properties of a file, in the Files tab on the right side of the window:

1. Select a file in the list.
2. Right-click the file and choose **File Properties**.



File properties

Filename: C:/Pirana/Pirana_Work/LocalRun3/Theophylli

Size: 4.5 kB

Last modified: Fri Oct 18 07:09:29 2019

Creator:

Description:

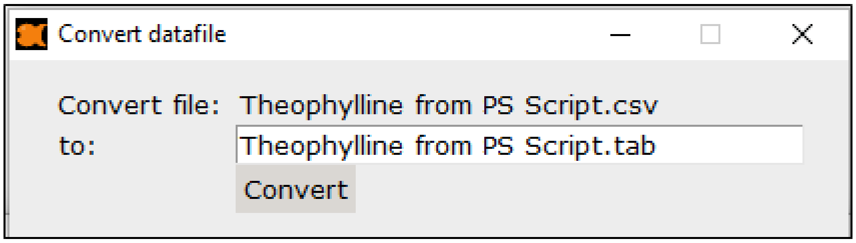
Notes:

3. In the dialog, modify the **Creator**, **Description**, or **Notes** as needed.
4. Press **Save and close**.

Convert between comma- and tab-separated format

To convert a csv formatted file to tab or a tab formatted file to csv, in the Files tab on the right side of the window:

1. Select a csv or tab formatted file in the list.
2. Right-click the file and choose **File actions > Convert CSV<->TAB**.


Convert datafile

Convert file: Theophylline from PS Script.csv

to: Theophylline from PS Script.tab

3. In the dialog, enter a name for the new file.
4. Press **Convert**.

Copy and move files

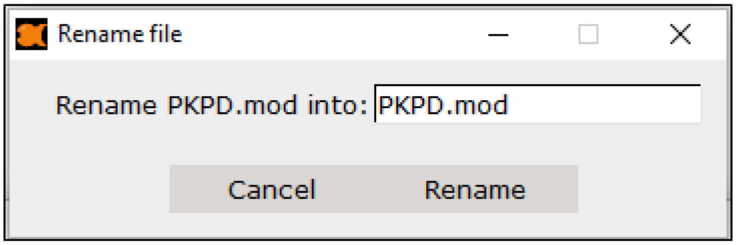
To copy or move a file to a subdirectory of the current folder, in the Files tab on the right side of the window:

1. Select one or more files in the list.
2. Right-click a selected file and choose **File actions > Copy/move file(s)**.
3. In the dialog, use the **Action** pulldown to indicate whether the file(s) should be copied or moved.
4. Select the subfolder in which to place the copied or moved file(s) from the **To subfolder** pulldown.
5. Press **Copy/move**.

Rename files

To rename a file, in the Files tab on the right side of the window:

1. Select the file in the list.
2. Right-click the file and choose **File actions > Rename file**.


Rename file

Rename PKPD.mod into: PKPD.mod

3. Enter the new name in the field and press **Rename**

Delete files

To delete files, in the Files tab on the right side of the window:

1. Select one or more files in the list.
2. Right-click a selected file and choose **File actions > Delete file(s)**.

3. In the dialog, verify the files to be deleted and press **Delete**.

Models

Models found in the project are listed in the model overview area, below the folders. By default, the models are ordered by run number. It is advised, but not mandatory, that models are named as a number (e.g., `001.mmdl`), or prepended with run (e.g., `run1.mod` or `run001.mod`). If you are working with NONMEM models, refer to “NONMEM template control file syntax” for additional NONMEM-specific information.

When models are double-clicked, the model is opened in the code-editor (if an editor is specified in the Software integration **Extensions** sub-tab in the *Settings* dialog or in Pirana’s built-in NM-TRAN editor.

The list of models in the overview area can be managed in the following ways:

- Condensed versus expanded mode
- List versus tree view
- Change the columns displayed
- Refresh model information
- Hide and unhide models
- Filtering model list
- Identify models and results by color
- Flag models and results

Pirana also provides functionality for performing the following model file tasks:

- Duplicate model files
  (Duplicate NLME models; for NONMEM models, click “Duplicate models”)
- Batch editing of models
- Open the input dataset
- Attach folders to selected model
- Rename model file
  (Rename NLME models; for NONMEM models, click “Rename a model file”)
- View difference with reference model
- Delete a model

Condensed versus expanded mode

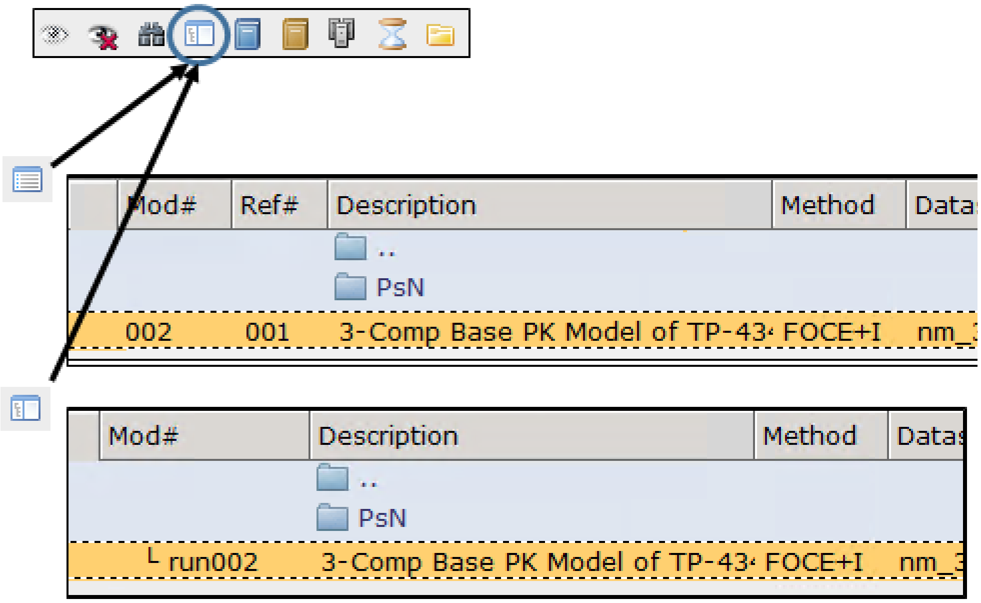
In ***condensed mode***, every model in the model overview list is shown as a single row in the table. The list can, however, also be shown in ***expanded mode***, which allows for longer model descriptions and notes in the overview (see “Add notes and view model properties”). Additionally, in the expanded mode, all estimation methods are shown, while in condensed mode only the last estimation method and associated OFV (Objective Function Value) are shown.

To switch between modes, click in the toolbar.
Or
Select the **View > Condensed/Expanded View** menu option.
Or
Right-click a model and select **Batch operations > Condensed/Expanded View**.

List versus tree view

Models are presented in ***list view*** by default. An alternate view mode is ***tree view***, in which model development is shown as a hierarchical tree.

To switch views, click in the toolbar. The icon will change to as an indication that you are in tree view mode.



	Mod#	Ref#	Description	Method	Data
			..		
			PsN		
	002	001	3-Comp Base PK Model of TP-43	FOCE+I	nm_

	Mod#	Description	Method	Data
		..		
		PsN		
	└ run002	3-Comp Base PK Model of TP-43	FOCE+I	nm_

The tree is built using parent/reference information included in the model files (see “Run records”). When creating models in Pirana, this information is added automatically and adheres to PsN’s run record syntax.

Change the columns displayed

Control the information presented in the model overview table by select the **View > Columns** menu and selecting/unselecting the columns in the list.

Refresh model information

Click in the toolbar to refresh the information for the selected model(s).

A refresh should be done when you make changes to models. Also when a run is finished, you should refresh to gather the results into Pirana.

Hide and unhide models

1. Select the model(s) to hide in the list.
2. Click in the toolbar.
   Or
   Select the **View > Hide model(s) from overview** menu option.
   Or
   Right-click the model and choose **Model > Hide model(s) from overview**.
   Or
   Right-click the model and choose **Batch operations > Hide model(s) from overview**.

When models are hidden, an indicator is added to the toolbar showing the number of hidden models.

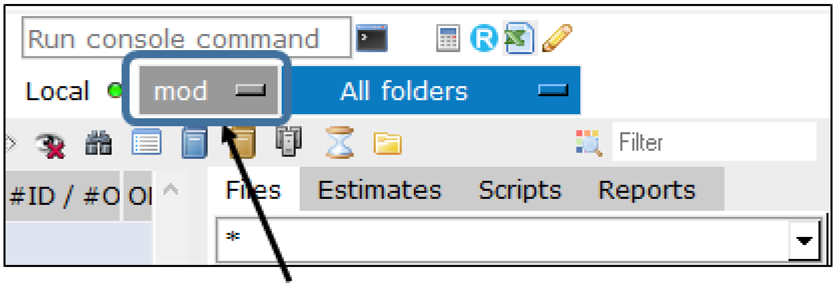
3. To unhide the models, click in the toolbar.
Or
Select the **View > Unhide models** menu option.
Or
Right-click a selected model and choose **Model > Unhide models**.
Or
Right-click a selected model and choose **Batch operations > Unhide models**.

Filtering model list

- Filter by file extension
- Filter by folder
- Filter by text
- Filter by color

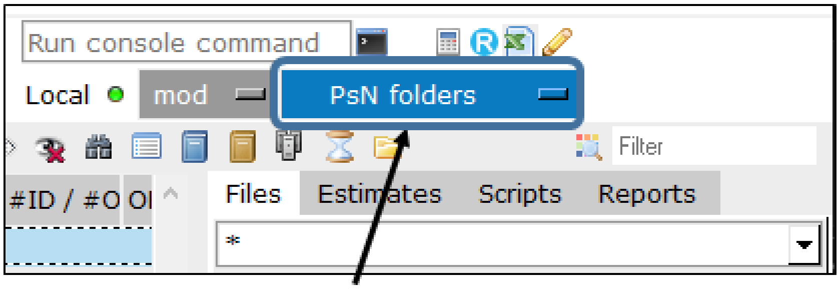
Filter by file extension

The model overview will only show models that have a file-extension corresponding to the file extension specified in the *Settings* dialog (**Miscellaneous settings** tab). If multiple file extensions were specified in the *Settings* dialog, use the extensions pull-down to filter the list.



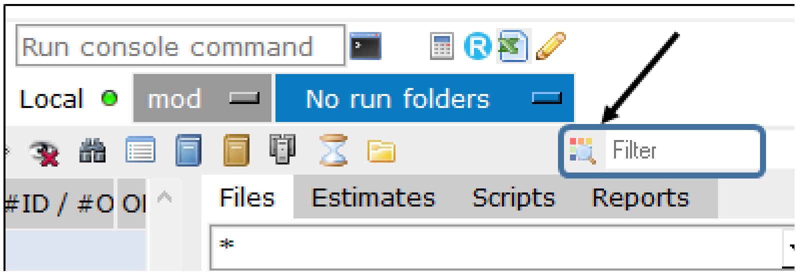
Filter by folder

Restrict the list of models in the model overview to only those in a specific folder by selecting the folder from the pull-down.



Filter by text

In the toolbar, begin typing in the **Filter** field.



Both folders and models will be filtered.

Filter by color

1. Click the ▦ icon (next to the **Filter** field) and choose **Show/Hide by color/Show All**.
2. If **Show/Hide by color** is selected, choose the color on which to filter.
3. Select **Show All** to remove any applied filtering by color.

See also "Identify models and results by color".

Identify models and results by color

To each model or result in Pirana, you can attach a color. The colors can indicate, e.g., key runs, good runs, or bad runs, but of course the meaning of the color-coding is all up to the user.

To assign a color to a model or result in the model overview:

1. Select the model or result in the list.
2. Click ▦ in the toolbar and select the desired color from the popup.
   Or
   Right-click the selected model or result and choose the desired color from the **Colors & flags** sub-menu.

Note: The color information is stored in a database file (`pirana.dir`), which is created automatically in each folder that holds models and is visited by Pirana. If you archive your projects manually, make sure to include these files as well.

Flag models and results

A flag can be attached to a model or result. As with colors, flags can indicate, e.g., key runs, good runs, or bad runs, but of course the meaning of the flags is all up to the user.

To assign a flag to a model or result in the model overview:

1. Select the model or result in the list.
2. Click 🚩 in the toolbar and select the desired flag from the popup.
   Or
   Right-click the selected model or result and choose the desired flag from the **Colors & flags** sub-menu.

Batch editing of models

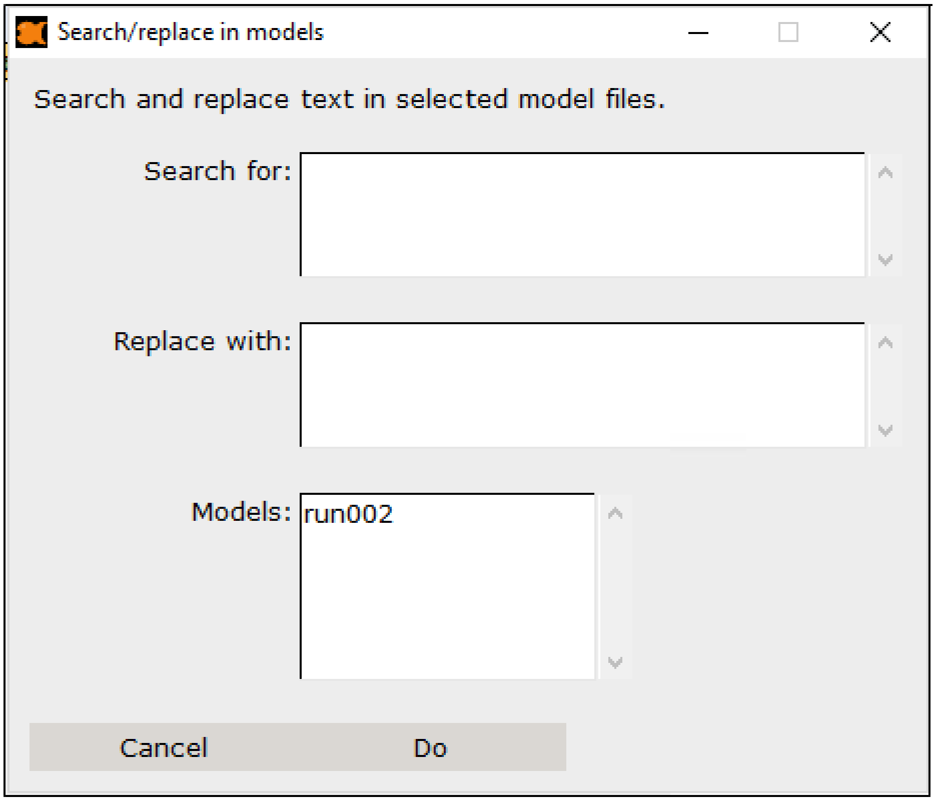
- Search and replace in models
- Change dataset
- Replace block in multiple models
- Add code to multiple models
- Add code to blocks in multiple models

Search and replace in models

Select the **Tools > Batch operations > Search and replace in models** menu option.
Or

Click in the toolbar and select **Search and replace in models** from the menu.

Replaces a given search text with another string or block of text in the selected models.



Search and replace text in selected model files.

Search for:

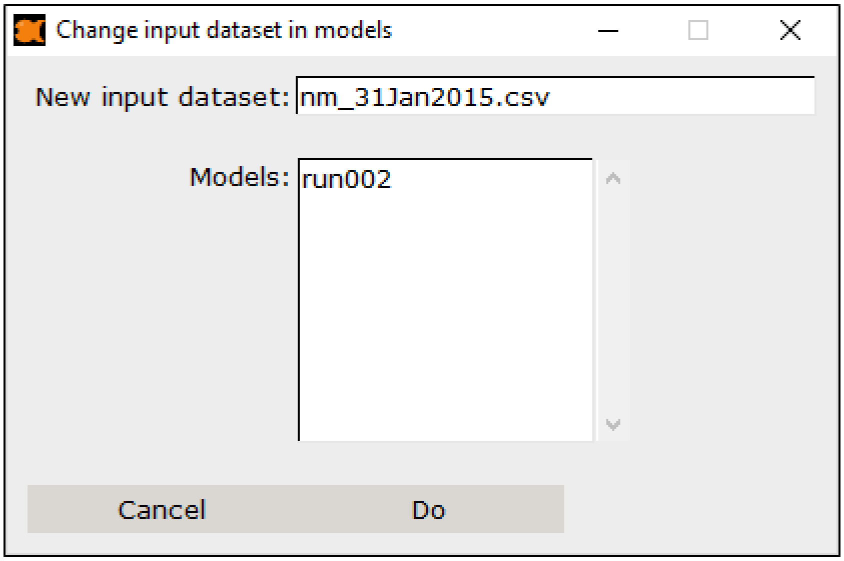
Replace with:

Models: run002

Change dataset

Select the **Tools > Batch operations > Change dataset** menu option.
Or

Click in the toolbar and select **Change dataset** from the menu.



Change input dataset in models

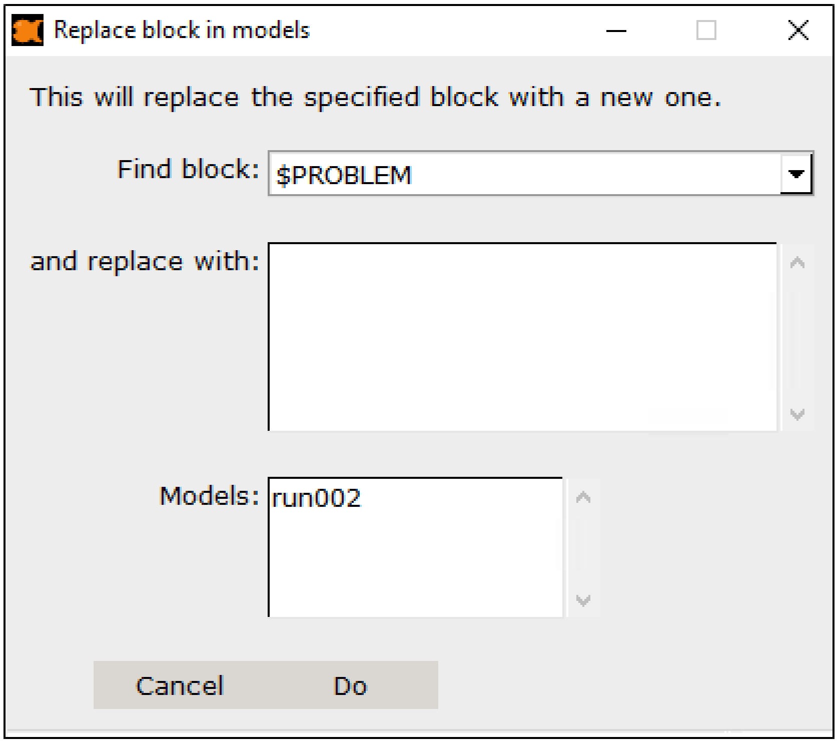
New input dataset: nm_31Jan2015.csv

Models: run002

Replace block in multiple models

Select the **Tools > Batch operations > Replace blocks** menu option.
Or

Click in the toolbar and select **Replace block** from the menu.



Replace block in models

This will replace the specified block with a new one.

Find block: $PROBLEM

and replace with:

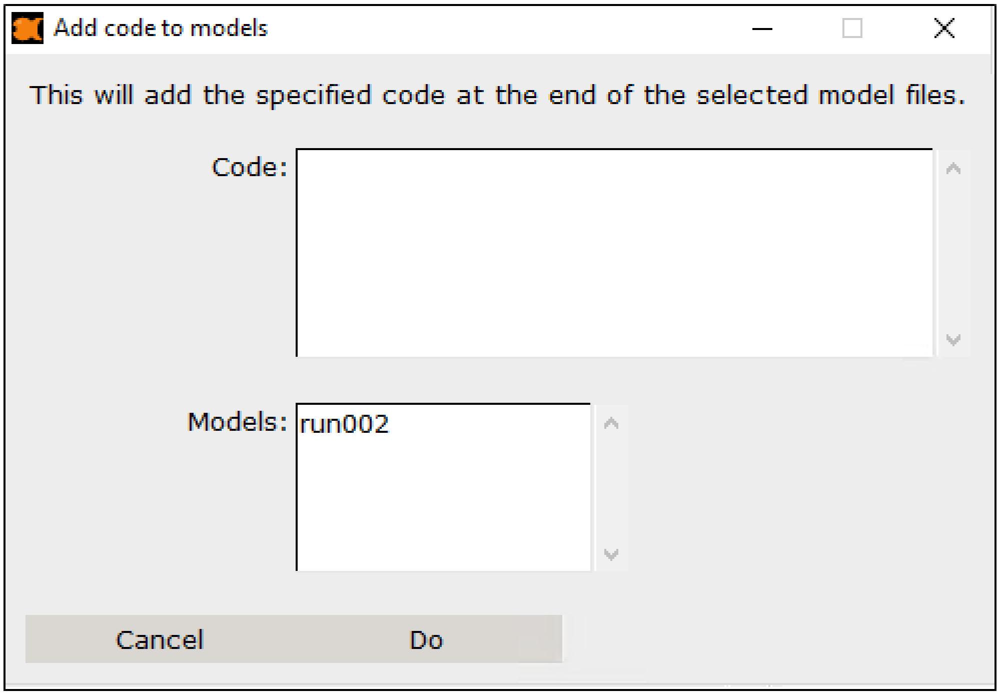
Models: run002

This function enables you to replace a whole block of code in selected model files. For example, replace the $DATA block if you want all model files to use a different data file or the $THETA block if you want to use other initial estimates.

Add code to multiple models

Select the **Tools > Batch operations > Add code to models** menu option.
Or

Click in the toolbar and select **Add code to models** from the menu.



Add code to models
This will add the specified code at the end of the selected model files.
Code:
Models:
Cancel
Do

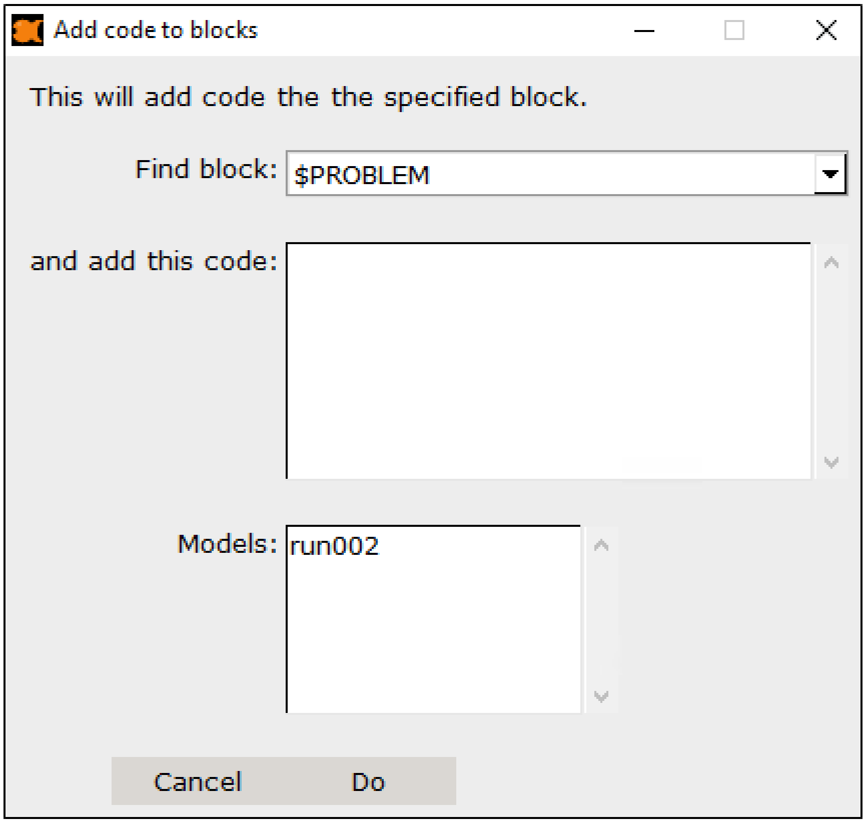
Models: run002

With this function, lines of code can be added at the end of selected models.

Add code to blocks in multiple models

Select the **Tools > Batch operations > Add code to blocks** menu option.
Or

Click in the toolbar and select **Add code to block** from the menu.



Add code to blocks

This will add code the the specified block.

Find block: $PROBLEM

and add this code:

Models: run002

With this function, lines of code can be added to a specific block in the selected models. The list of blocks in the **Find block** pulldown will differ if you are looking at NLME models or NONMEM models.

Open the input dataset

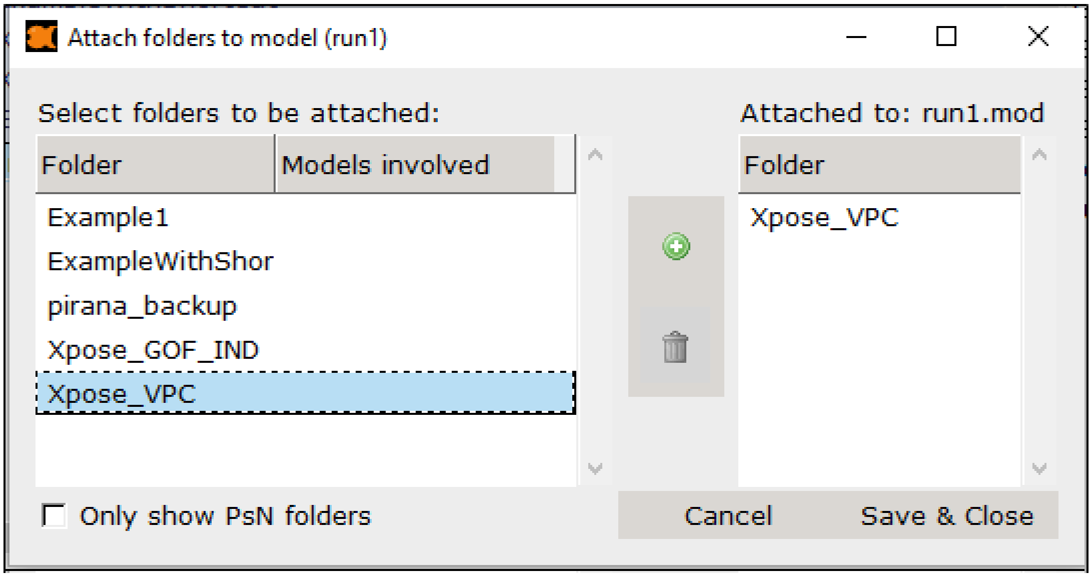
1. Select the model in the list.
2. Right-click the selected model and choose **Model > Open input dataset** from the right-click menu.

The input dataset associated with the model will open as an Excel spreadsheet.

Attach folders to selected model

To attach one or more folders to a model. Attaching a folder creates a connection to the model, e.g., attached folders will be highlighted in the list when the model is selected. Later, the folder can be handled in conjunction with the model. For example, when the model is archived or deleted, the attached folders are also archived or deleted.

1. Click 🗂 in the toolbar.
   Or
   Select **Tools > Attach folders to selected models**.
   Or
   Right-click a model and select **File action > Attach folders to model(s)**.



Attach folders to model (run1)

Select folders to be attached:

Folder	Models involved
Example1	
ExampleWithShor	
pirana_backup	
Xpose_GOF_IND	
Xpose_VPC	

Attached to: run1.mod

Folder
Xpose_VPC

□ Only show PsN folders

2. In the dialog, select one or more folders to attach.
   For NONMEM models, check the **Only show PsN folders** to filter the list on the left.
   For NLME models, the checkbox is labeled **Only show NLME folders**.

3. Click ⊕ to add the folders to the list on the right.

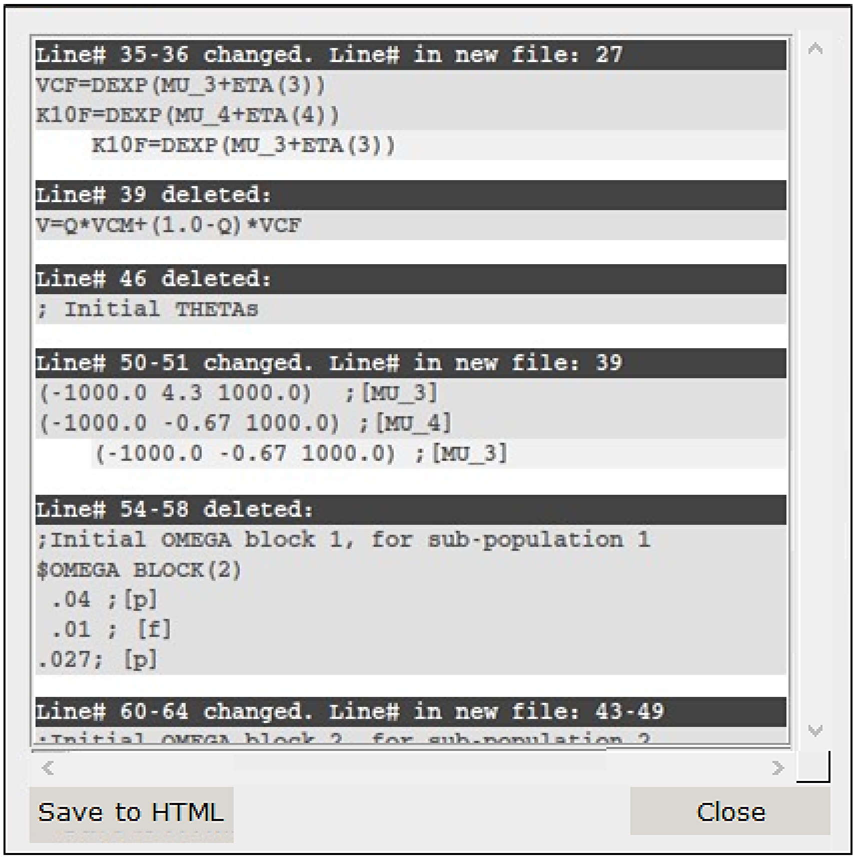
4. Use the 🗑 to remove a folder from the list on the right.

5. Press **Save & Close**.

View difference with reference model

Pirana provides a tool to show code differences, similar to the diff functionality on Unix systems. If one model is selected and the `diff` tool is activated, Pirana will show the difference between that model and the reference mode (if specified). If two models are selected, Pirana will show the code differences between the two selected models.

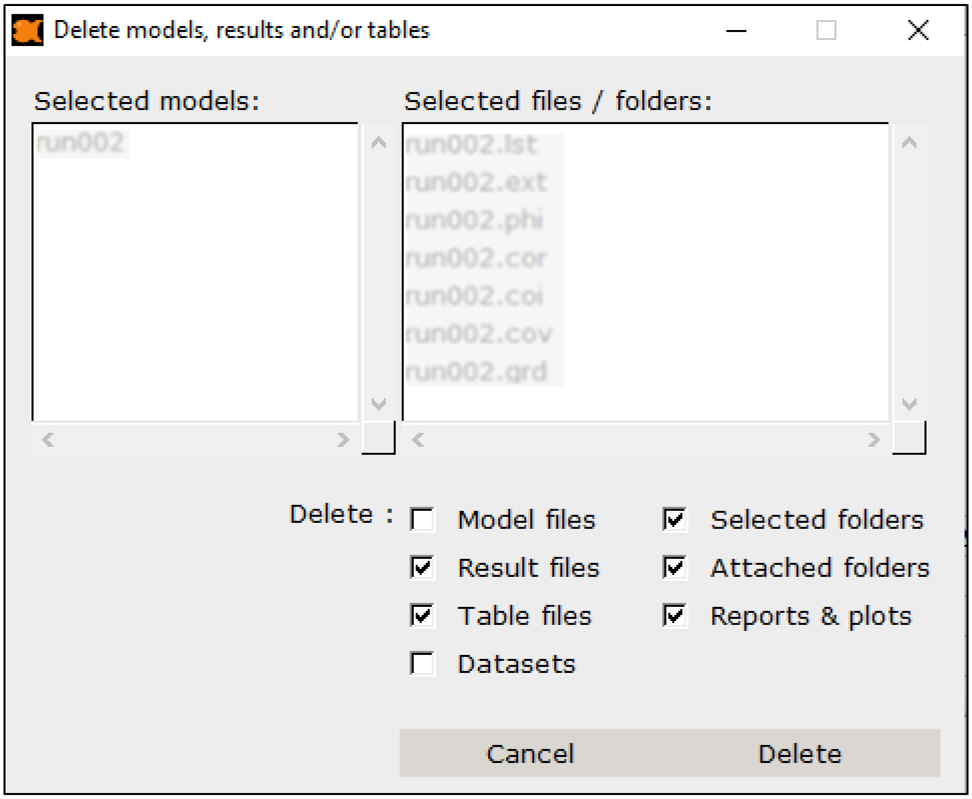
1. Select a model in the list.
2. Right-click the selected model and choose **Model > Code difference between models** from the menu.
   Or
   Select **Results > Run reports > Difference with reference model**.



```
Line# 35-36 changed. Line# in new file: 27
VCF=DEXP(MU_3+ETA(3))
K10F=DEXP(MU_4+ETA(4))
    K10F=DEXP(MU_3+ETA(3))

Line# 39 deleted:
V=Q*VCM+(1.0-Q)*VCF

Line# 46 deleted:
; Initial THETAs

Line# 50-51 changed. Line# in new file: 39
(-1000.0 4.3 1000.0)  ;[MU_3]
(-1000.0 -0.67 1000.0) ;[MU_4]
    (-1000.0 -0.67 1000.0) ;[MU_3]

Line# 54-58 deleted:
;Initial OMEGA block 1, for sub-population 1
$OMEGA BLOCK(2)
 .04 ;[p]
 .01 ;  [f]
.027;  [p]

Line# 60-64 changed. Line# in new file: 43-49
;Initial OMEGA block 2, for sub-population 2
```

Delete a model

1. Select a model in the list.
2. Right-click the selected model and choose **File actions > Delete** from the menu.
   Or
   Select **Models > Delete model(s)** from the main menu.
   Or
   Click 🗑 in the toolbar.
   Or
   Select the model in the list and press the keyboard button **DELETE**.



Delete models, results and/or tables

Selected models:
run002

Selected files / folders:
run002.lst
run002.ext
run002.phi
run002.cor
run002.coi
run002.cov
run002.grd

Delete :
□ Model files
☑ Result files
☑ Table files
□ Datasets
☑ Selected folders
☑ Attached folders
☑ Reports & plots

3. In the dialog, check the boxes for the items associated with the models that you wish to delete.

Note: If you have selected one or more folders in the main overview to be deleted, the **Folder** option should be checked to actually delete these as well.

Data Inspection

- Data Inspector
- Correlation calculator
- Check dataset

Data Inspector

Pirana is able to construct scatter plots using the built-in DataInspector for quickly inspecting goodness-of-fit, covariate relationships, distribution of etas, performing data checkout, etc. The DataInspector shows all the columns present in the dataset, which can be mapped to the X- or Y-axis (shown in following figure).

For a data or table file:

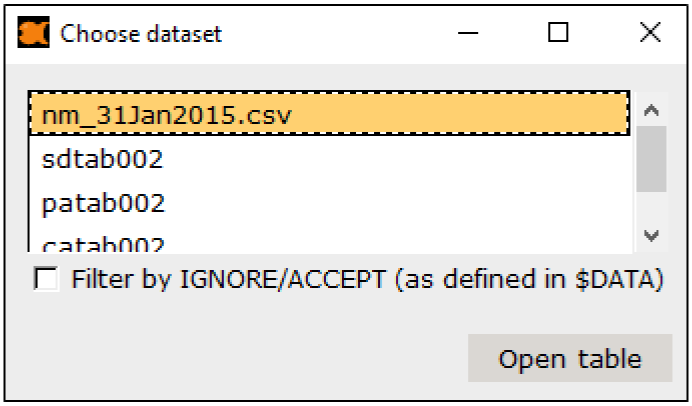
1. In the Files tab on the right, select a dataset or table file.
2. Right-click the selected file and choose **Open in > DataInspector**.

Or

1. From the Estimates tab on the right, click 📈 in the toolbar.

For a model:

1. In the model overview list, select a model that contains $TABLE records.
2. Select **Results > Open DataInspector** from the main menu.
   Or
   Right-click the selected model and choose **Model > Open DataInspector**.
   Or
   Click 📈 in the toolbar.
3. If multiple datasets are available for the model, select the one to inspect in the displayed dialog.



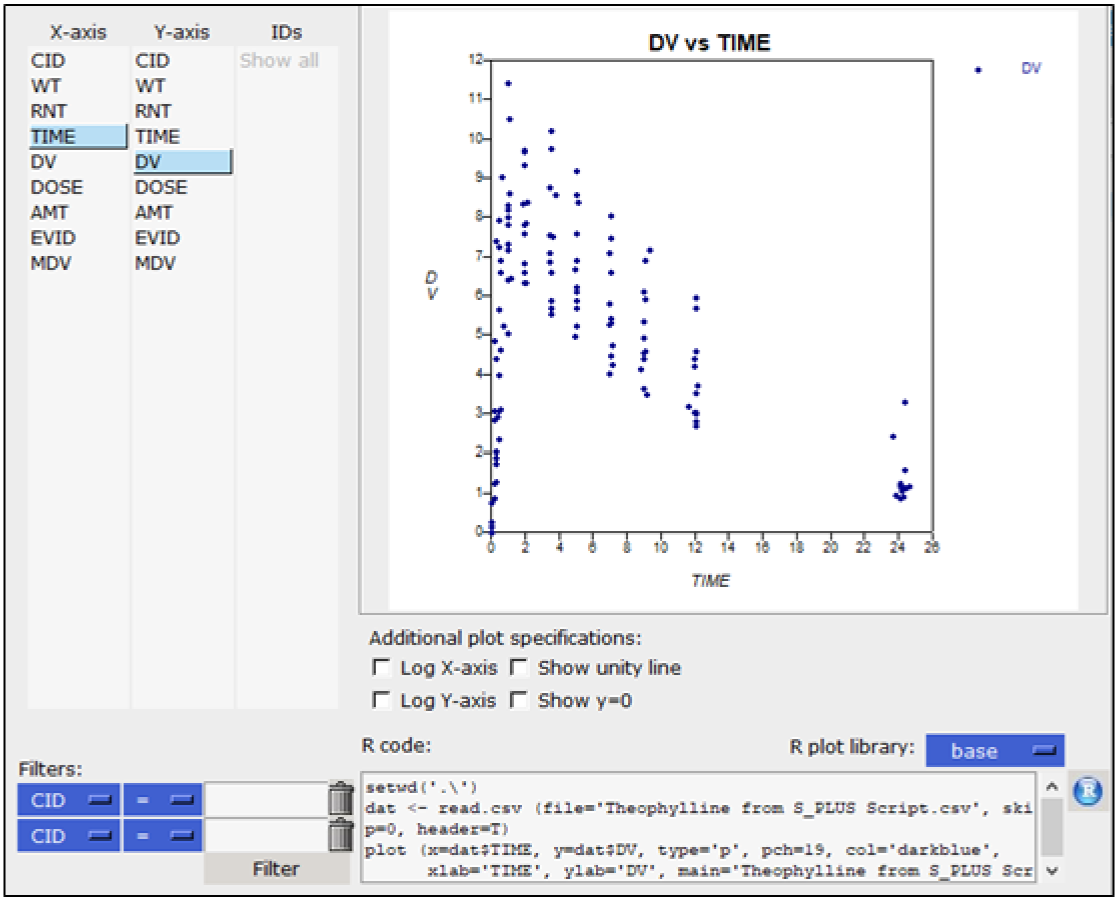
Choose dataset

nm_31Jan2015.csv
sdtab002
patab002
catab002

☐ Filter by IGNORE/ACCEPT (as defined in $DATA)

Open table

   – Check the **Filter by IGNORE/ACCEPT** box to use the filtering defined in the dataset. If unchecked, all information in the dataset will be loaded into the DataInspector.
   – Press **Open table**.



X-axis	Y-axis	IDs
CID	CID	Show all
WT	WT	
RNT	RNT	
TIME	TIME	
DV	DV	
DOSE	DOSE	
AMT	AMT	
EVID	EVID	
MDV	MDV	

Additional plot specifications:

☐ Log X-axis ☐ Show unity line

☐ Log Y-axis ☐ Show y=0

R code:

R plot library: base

Filters:

CID =

CID =

Filter

```
setwd('.\')
dat <- read.csv (file='Theophylline from S_PLUS Script.csv', ski
p=0, header=T)
plot (x=dat$TIME, y=dat$DV, type='p', pch=19, col='darkblue',
      xlab='TIME', ylab='DV', main='Theophylline from S_PLUS Scr
```

Hovering over a point in the graph will display the value of that data point.

4. Select a column name from the **X-axis** and **Y-axis** lists to switch the information that is plotted in the graph.
5. Select an ID(s) from the **IDs** list to only include data associated with the selected ID(s) in the graph. (There is no ID column in the source for the above image.)
6. Use the **Filters** pull-downs to define additional filter criteria.
   - Select the dataset column name from the first pull-down.
   - Select the operator from the second pull-down.
   - Enter a value in the field.
   - Click 🗑 to clear the corresponding criterion.
   - Press the **Filter** button to add another criterion.
7. Use the **Additional plot specifications** checkboxes to change the scaling of the axes (**Log X-axis** and **Log Y-axis**), display a unity line (**Show unity line**), and/or start the Y axis at the origin (**Show y=0**).
8. Select the **R plot library** to use from the pull-down.
9. In the R code field, edit the R code for generating the plot in R, as needed.
10. Multiple Y values can be plotted by holding the **Control-** or **shift**-key and selecting multiple (up to three) data columns in the DataInspector.
11. Inside the plot, regions of interest may be selected, which are then zoomed.

12. Double-clicking inside the plot region changes back to the previous view.
13. Using the DataInspector tools, data can be filtered, which can be useful, to show only data for one patient, or groups of patients or covariates in the plots.
14. Hover the cursor over a note to see additional model information.

    In the text-box below the plot, code is generated that recreates the same graph in R, either in `base`, `lattice`, or `ggplot2`. This code can be used as a starting point for the generation of plots for manuscripts or reports.

15. Click R to open the script file within R-GUI or RStudio.

    You can execute the script as-is or make modifications to the code and execute. Copying the model text and pasting it into an R-GUI or RStudio script file is also acceptable.

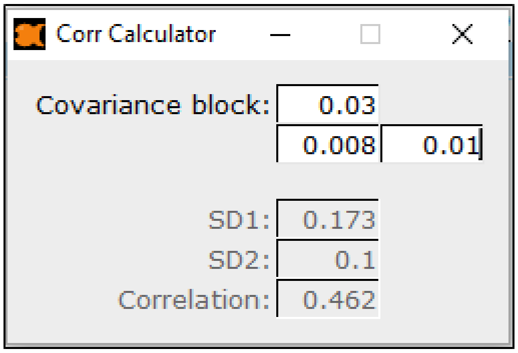
Correlation calculator

The *Correlation Calculator* re-calculates a covariance to a correlation on the SD-scale. The formula for correlation that is used is:

$$\rho_{i,j} = \frac{\omega^{2}{}_{i,j}}{\omega_{i,i}\cdot\omega_{j,j}}$$

with ρ specifying the correlation between two elements (i,j) in a matrix, and ω specifying elements of Ω or Σ.

1. Select **Tools > Correlation calculator** from the main menu.



Covariance block: 0.03
0.008 0.01

SD1: 0.173
SD2: 0.1
Correlation: 0.462

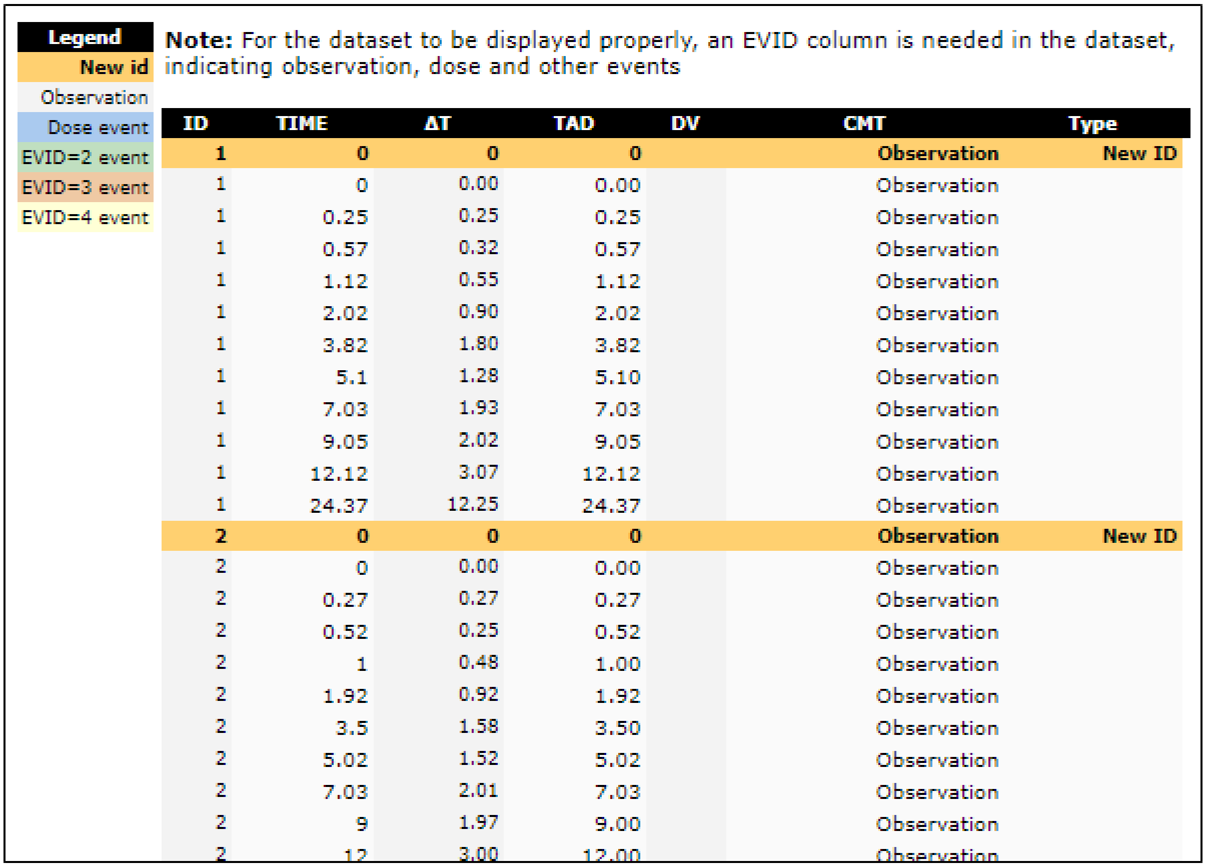
2. In the dialog, enter covariance values in the appropriate fields.

Resulting standard deviations for each covariant, and the correlation are automatically computed and displayed.

Check dataset

1. Select **csv** from the pulldown at the top of the Files tab on the right side of the window.
2. Select a dataset file from the list.
3. Right-click the selected file and choose **Check Dataset** from the menu.

This will create (and open) an HTML file which displays a selected dataset using separate colors for different event types. Thus showing the dataset in a slightly more convenient format for manual inspection than in a spreadsheet. The function needs at least the ID, TIME, and EVID columns in the dataset to work properly.



Legend
New id
Observation
Dose event
EVID=2 event
EVID=3 event
EVID=4 event

Note: For the dataset to be displayed properly, an EVID column is needed in the dataset, indicating observation, dose and other events

ID	TIME	ΔT	TAD	DV	CMT	Type
1	**0**	**0**	**0**		**Observation**	**New ID**
1	0	0.00	0.00		Observation	
1	0.25	0.25	0.25		Observation	
1	0.57	0.32	0.57		Observation	
1	1.12	0.55	1.12		Observation	
1	2.02	0.90	2.02		Observation	
1	3.82	1.80	3.82		Observation	
1	5.1	1.28	5.10		Observation	
1	7.03	1.93	7.03		Observation	
1	9.05	2.02	9.05		Observation	
1	12.12	3.07	12.12		Observation	
1	24.37	12.25	24.37		Observation	
2	**0**	**0**	**0**		**Observation**	**New ID**
2	0	0.00	0.00		Observation	
2	0.27	0.27	0.27		Observation	
2	0.52	0.25	0.52		Observation	
2	1	0.48	1.00		Observation	
2	1.92	0.92	1.92		Observation	
2	3.5	1.58	3.50		Observation	
2	5.02	1.52	5.02		Observation	
2	7.03	2.01	7.03		Observation	
2	9	1.97	9.00		Observation	
2	12	3.00	12.00		Observation	

Analyzing Results and Output

After a model has been run/executed, and the folder is refreshed, Pirana will show the main results of the run in the main window. It will show the OFV, the difference in OFV with the reference model (if specified), the number of significant digits, and some information about the estimation:

S = successful minimization

R = estimation ended with rounding errors

C = successful covariance step

M = unsuccessful covariance step due to matrix singularity

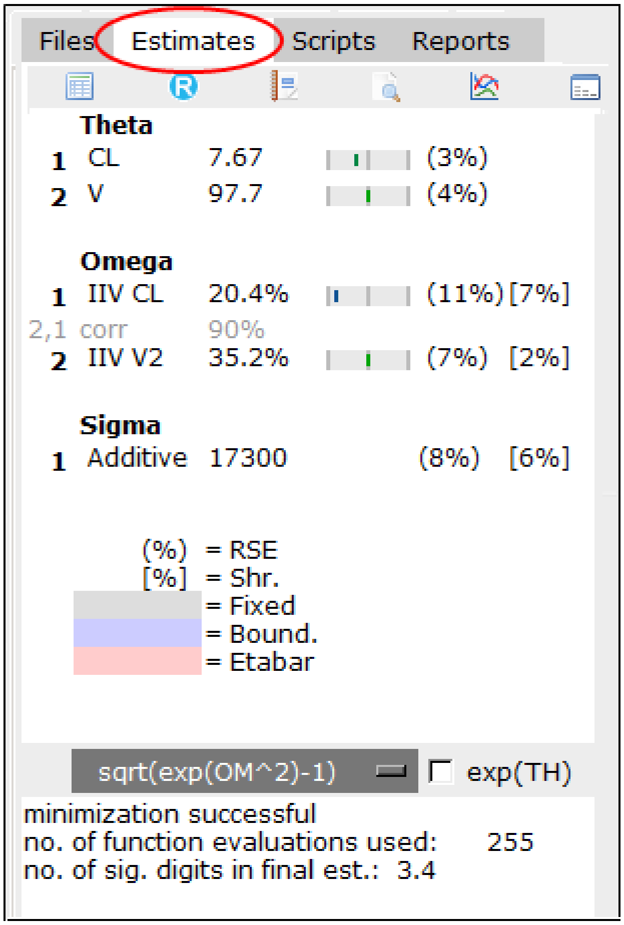
B = boundary problem

Additional information on the following topics is available:

- View and compare parameter estimates
- View execution log
- Calculate AIC/BIC
- Matrices
- R scripts for graphs and file processing

View and compare parameter estimates

A list of parameter estimates is available in the **Estimates** tab in the right section of the *Pirana* window.



```
     Theta
1    CL         7.67            (3%)
2    V          97.7            (4%)

     Omega
1    IIV CL     20.4%           (11%) [7%]
2,1  corr       90%
2    IIV V2     35.2%           (7%)  [2%]

     Sigma
1    Additive   17300           (8%)  [6%]

     (%)  = RSE
     [%]  = Shr.
     ▨    = Fixed
     🟦   = Bound.
     🟥   = Etabar
```

sqrt(exp(OM^2)-1) ☐ exp(TH)

```
minimization successful
no. of function evaluations used:     255
no. of sig. digits in final est.:  3.4
```

Through the use of brackets and color additional information is presented in the **Estimates** tab.

(RSE for parameters is shown in round brackets)

[Shrinkage for the random effects is shown in square brackets]

Parameters where the final gradients were zero have red text

Parameters were fixed have a gray background

Parameters where boundaries were encountered have a blue background

Parameters where the mean of eta-distribution was significantly different from zero include the etabar with a red background

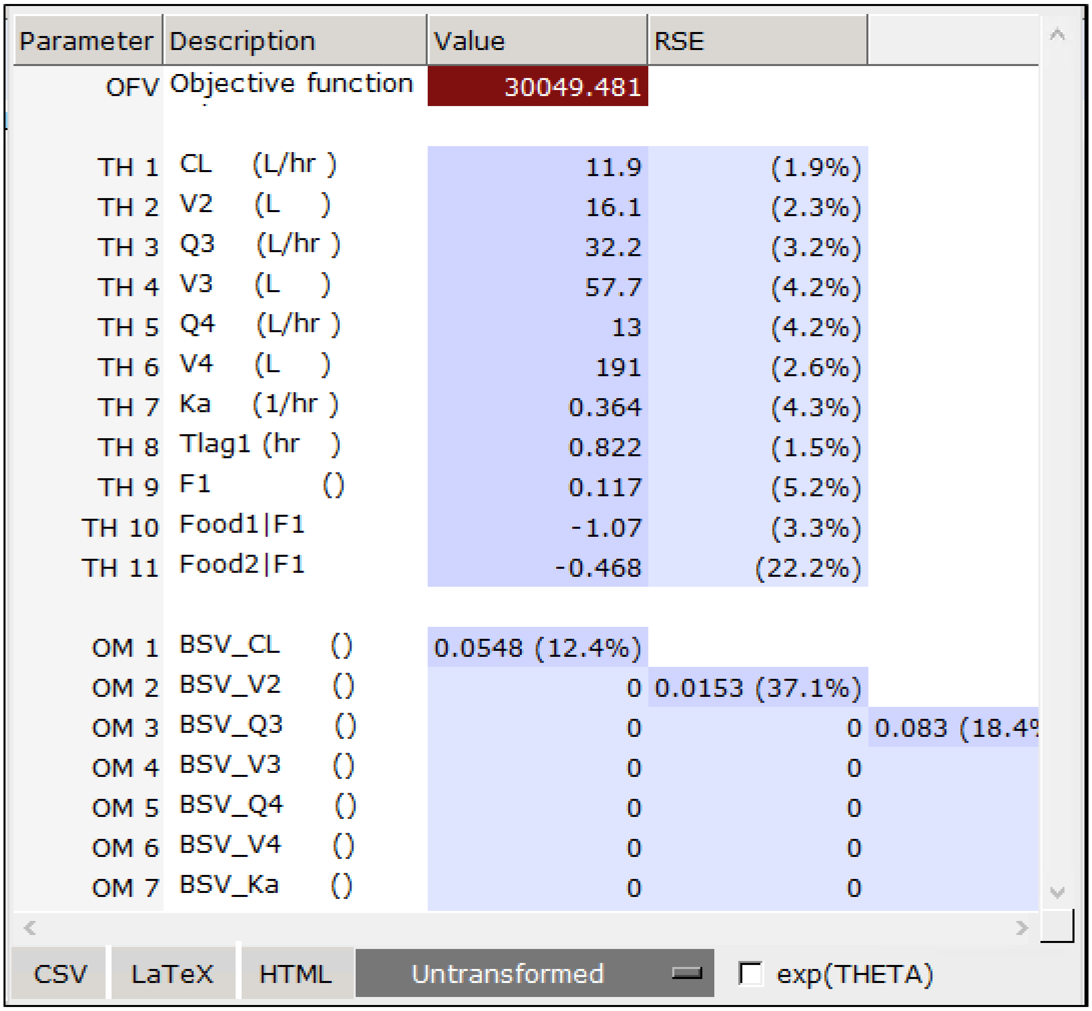
Using the **Estimates** tab toolbar, you can easily access the following tools and functionality:

- : Display the *Parameter Estimates* dialog, which gives a more detailed list of estimates.
- : Export the estimates using R.
- : Generate report.
- : View the NM output file.

- : Open dataset/output files in DataInspector
- : View matrix. Select type of matrix from the pop-up menu (see “Matrices” for more information).

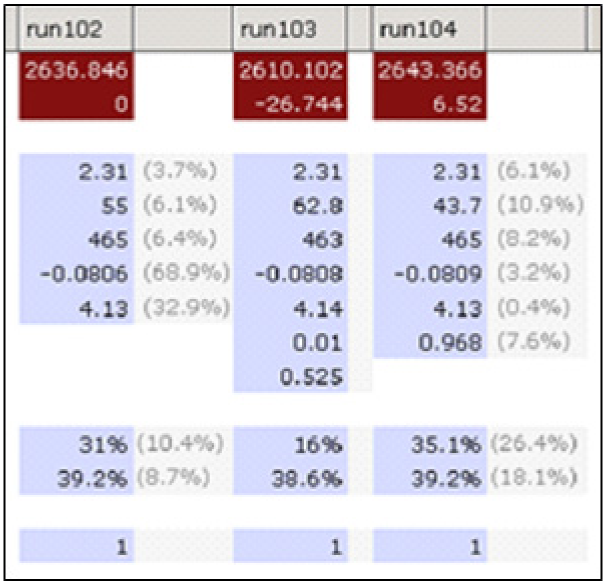
The more detailed *Parameter Estimates* dialog can be accessed using the any of the following methods:

- Click in the Pirana toolbar.
- Click in the Estimates tab toolbar.
- Select **Results > Show/compare parameter estimates**.
- Select **View > Show parameter estimates**.
- Right-click a selected model in the model overview list and choose **Model > Parameter estimates** from the menu.
- Right-click a selected model in the model overview list and choose **Batch operations > Show parameter estimates** from the menu.



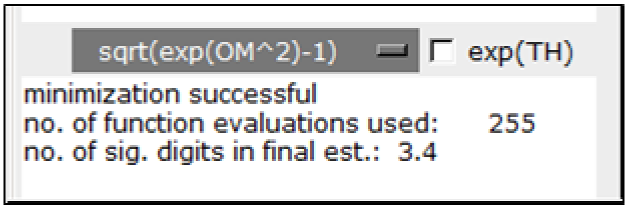
Parameter	Description	Value	RSE	
OFV	Objective function	30049.481		
TH 1	CL (L/hr)	11.9	(1.9%)	
TH 2	V2 (L)	16.1	(2.3%)	
TH 3	Q3 (L/hr)	32.2	(3.2%)	
TH 4	V3 (L)	57.7	(4.2%)	
TH 5	Q4 (L/hr)	13	(4.2%)	
TH 6	V4 (L)	191	(2.6%)	
TH 7	Ka (1/hr)	0.364	(4.3%)	
TH 8	Tlag1 (hr)	0.822	(1.5%)	
TH 9	F1 ()	0.117	(5.2%)	
TH 10	Food1|F1	-1.07	(3.3%)	
TH 11	Food2|F1	-0.468	(22.2%)	
OM 1	BSV_CL ()	0.0548 (12.4%)		
OM 2	BSV_V2 ()	0	0.0153 (37.1%)	
OM 3	BSV_Q3 ()	0	0	0.083 (18.4
OM 4	BSV_V3 ()	0	0	
OM 5	BSV_Q4 ()	0	0	
OM 6	BSV_V4 ()	0	0	
OM 7	BSV_Ka ()	0	0	

If multiple models/runs are selected when the *Parameter Estimates* dialog is opened, the parameter estimates for all selected runs are presented, side-by-side, facilitating comparison.



run102		run103	run104	
2636.846		2610.102	2643.366	
0		-26.744	6.52	
2.31	(3.7%)	2.31	2.31	(6.1%)
55	(6.1%)	62.8	43.7	(10.9%)
465	(6.4%)	463	465	(8.2%)
-0.0806	(68.9%)	-0.0808	-0.0809	(3.2%)
4.13	(32.9%)	4.14	4.13	(0.4%)
		0.01	0.968	(7.6%)
		0.525		
31%	(10.4%)	16%	35.1%	(26.4%)
39.2%	(8.7%)	38.6%	39.2%	(18.1%)
1		1	1	

The pull-down menu offers different choices for transforming the variances. Options include: **Untransformed**, (**sqrt(OM^2**), and (**sqrt(exp(OM^2)-1**).



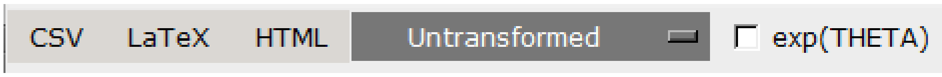
minimization successful
no. of function evaluations used: 255
no. of sig. digits in final est.: 3.4

Check the **exp(THETA)** box to back transform thetas that have been log-transformed in model code.

Note: Pirana will automatically back-transform when the parameter description includes the text “[log]”, e.g.:

```
#THETA
(0.,1.6,4.) ;KA [log]
```

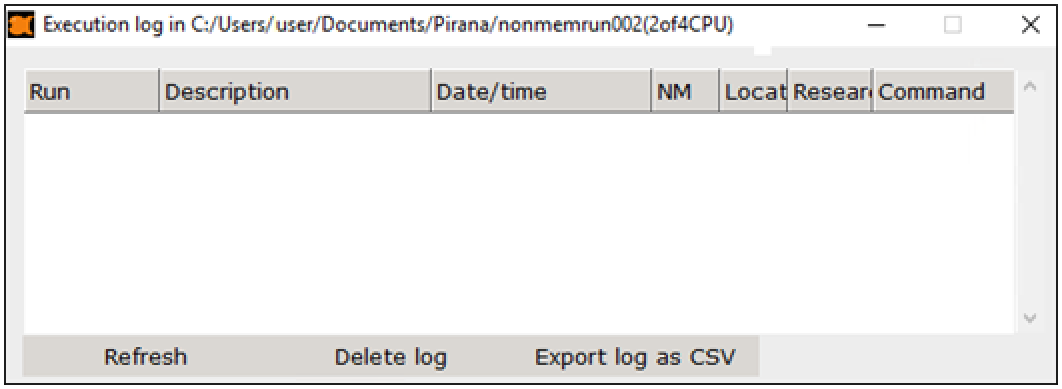
From the dialog, convert results into **CSV** format, **LaTeX**, or **HTML** for reports or further analysis using the buttons along the bottom.



View execution log

Each project has its own log file, and each execution is saved to that log. To view the log:

1. Select 📘 next to the **Project** pulldown in the *Pirana* window.
   Or
   Select **View > Execution log** from the main menu.
   Or
   Right-click a selected model and choose **Batch operations > Execution log**.



Execution log in C:/Users/user/Documents/Pirana/nonmemrun002(2of4CPU)

2. Click **Refresh** to reload the log file.
3. Click **Delete log** to delete the file.
4. Click **Export log as CSV** to create a CSV formatted version of the log file.

Calculate AIC/BIC

Pirana can calculate the Akaike Information Criterion and the Bayesian Information Criterion. These criteria are defined as follows:

$$AIC = 2 \cdot k - 2 \cdot \ln(L) \qquad (1)$$

$$BIC = -2 \cdot \ln(L) + k \cdot \ln(n) \qquad (2)$$

with

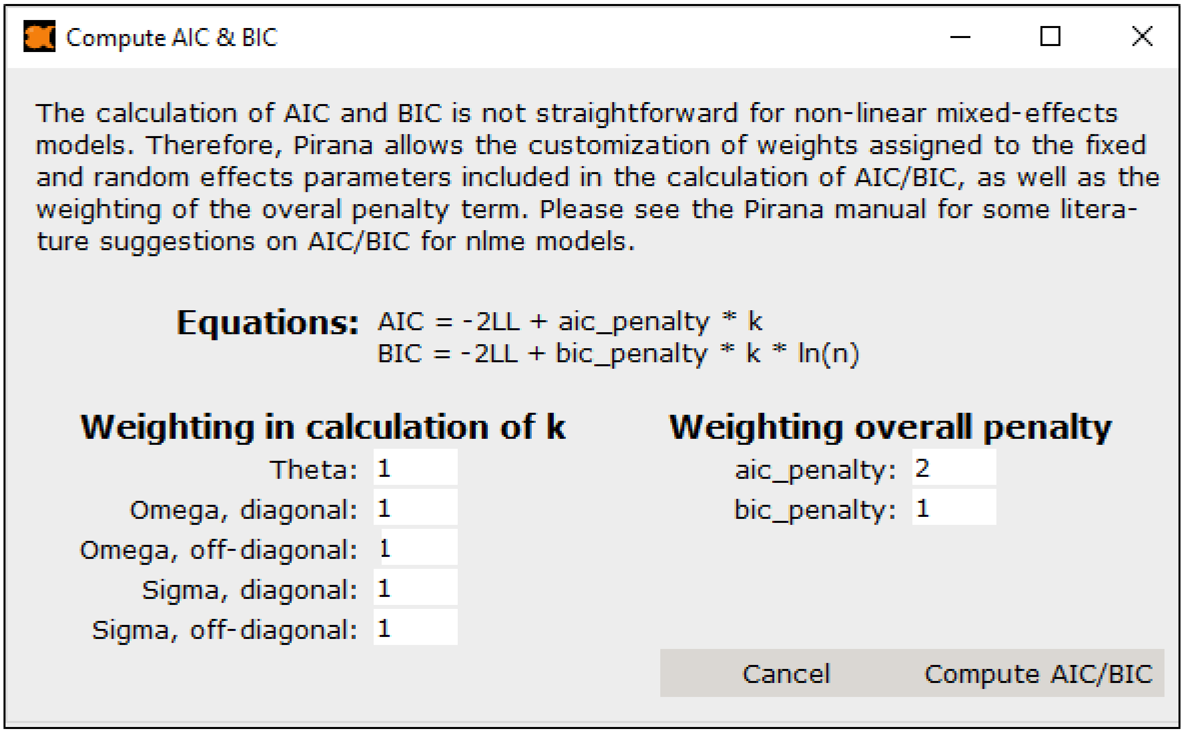
k = the number of parameters in the model,

L = the maximized value of the likelihood function, and

n = the number of observations in the dataset used in fitting the model.

The calculation of these criteria is, however, not so straightforward for non-linear mixed-effects models, and the weights/penalties applied to parts of the equation can be different in different circumstances. Pirana allows the penalties to be changed when it calculates the AIC/BIC.

1. Select the model in the list.
2. Click ⭐.
   Or
   Right-click the selected model and choose **Model > Comput AIC & BIC** from the menu.



Compute AIC & BIC

The calculation of AIC and BIC is not straightforward for non-linear mixed-effects models. Therefore, Pirana allows the customization of weights assigned to the fixed and random effects parameters included in the calculation of AIC/BIC, as well as the weighting of the overal penalty term. Please see the Pirana manual for some literature suggestions on AIC/BIC for nlme models.

Equations: AIC = -2LL + aic_penalty * k
BIC = -2LL + bic_penalty * k * ln(n)

Weighting in calculation of k

Theta: 1
Omega, diagonal: 1
Omega, off-diagonal: 1
Sigma, diagonal: 1
Sigma, off-diagonal: 1

Weighting overall penalty

aic_penalty: 2
bic_penalty: 1

3. Adjust the weightings and penalties for the various parts of the calculation as needed by typing directly in each field.
4. Click **Compute AIC/BIC**.

Some references to AIC and BIC literature are listed below.

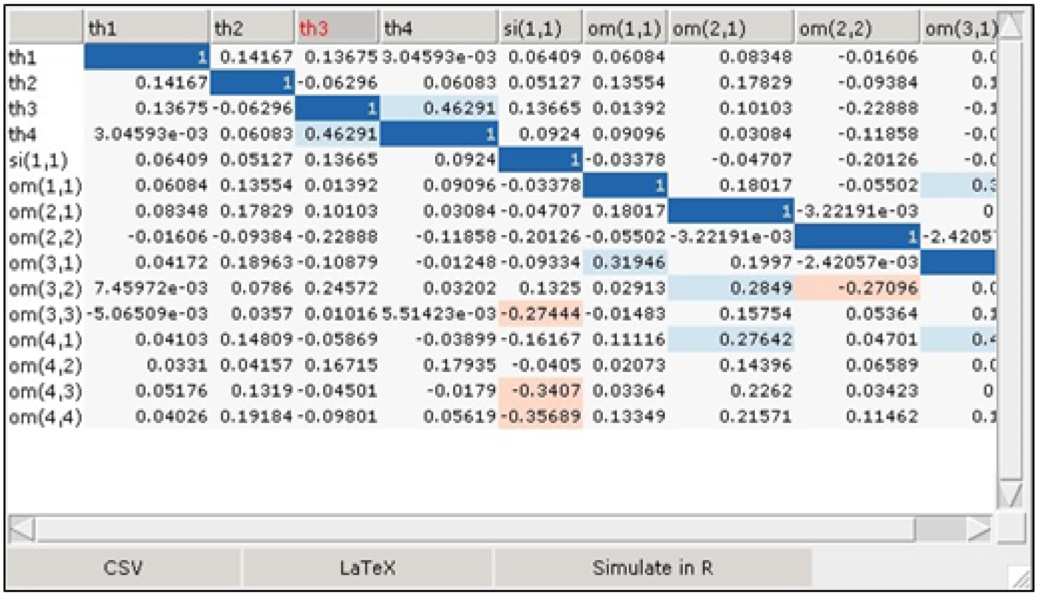
- Vaida and Blanchard (2005). Conditional Akaike information for mixed-effects models. *Biometrika* 92(2): 351-370.
- Liang, et al (2008). A note on conditional aic for linear mixed-effects models. *Biometrika* 95(3): 773-778.
- Hodges and Sargent (2001). Counting degrees of freedom in hierarchical and other richly-parameterized models. *Biometrika* 88(2): 367-379.
- Donohue et al. (2011). Conditional Akaike information under generalized linear and proportional hazards mixed models. *Biometrika* 98(3): 685-700.
- Delattre et al. BIC selection procedures in mixed effects models http://hal.inria.fr/docs/00/69/64/35/PDF/RR-7948.pdf.

Matrices

Pirana can automatically extract the covariance, correlation and inverse covariance matrices from a NONMEM 7+ run (`cor/cov/coi` files), and show them in a spreadsheet-like window. These can then also be automatically exported to an R object for simulation purposes using the **Simulate in R** button.

1. Select the model in the list.
2. In the Estimates tab, click ▤ in the toolbar.
   Or
   Right-click the selected model and choose one of the matrix types from the **Model > Matrices** sub-menu.

An example **Correlation Matrix** is shown below.



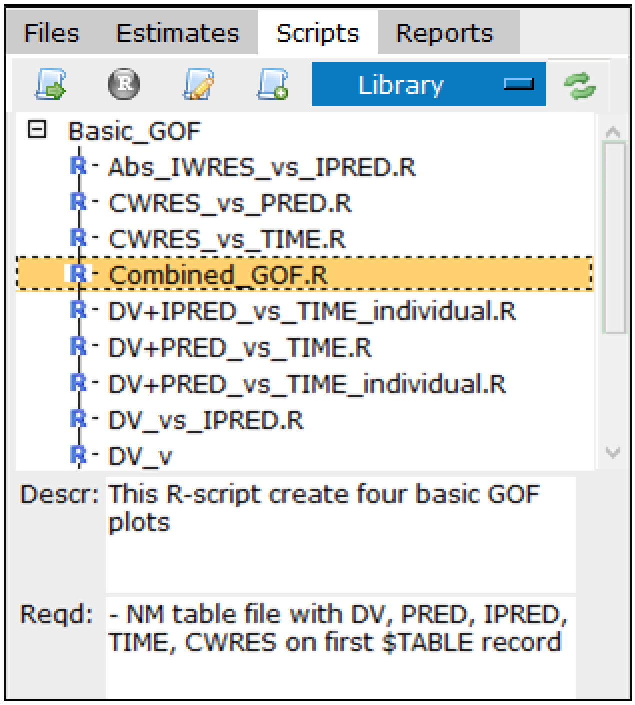
	th1	th2	th3	th4	si(1,1)	om(1,1)	om(2,1)	om(2,2)	om(3,1)
th1	1	0.14167	0.13675	3.04593e-03	0.06409	0.06084	0.08348	-0.01606	0.0
th2	0.14167	1	-0.06296	0.06083	0.05127	0.13554	0.17829	-0.09384	0.1
th3	0.13675	-0.06296	1	0.46291	0.13665	0.01392	0.10103	-0.22888	-0.1
th4	3.04593e-03	0.06083	0.46291	1	0.0924	0.09096	0.03084	-0.11858	-0.0
si(1,1)	0.06409	0.05127	0.13665	0.0924	1	-0.03378	-0.04707	-0.20126	-0.0
om(1,1)	0.06084	0.13554	0.01392	0.09096	-0.03378	1	0.18017	-0.05502	0.3
om(2,1)	0.08348	0.17829	0.10103	0.03084	-0.04707	0.18017	1	-3.22191e-03	0
om(2,2)	-0.01606	-0.09384	-0.22888	-0.11858	-0.20126	-0.05502	-3.22191e-03	1	-2.4205
om(3,1)	0.04172	0.18963	-0.10879	-0.01248	-0.09334	0.31946	0.1997	-2.42057e-03	
om(3,2)	7.45972e-03	0.0786	0.24572	0.03202	0.1325	0.02913	0.2849	-0.27096	0.0
om(3,3)	-5.06509e-03	0.0357	0.01016	5.51423e-03	-0.27444	-0.01483	0.15754	0.05364	0.1
om(4,1)	0.04103	0.14809	-0.05869	-0.03899	-0.16167	0.11116	0.27642	0.04701	0.4
om(4,2)	0.0331	0.04157	0.16715	0.17935	-0.0405	0.02073	0.14396	0.06589	0.0
om(4,3)	0.05176	0.1319	-0.04501	-0.0179	-0.3407	0.03364	0.2262	0.03423	0
om(4,4)	0.04026	0.19184	-0.09801	0.05619	-0.35689	0.13349	0.21571	0.11462	0.1

R scripts for graphs and file processing

- Interactive scripts
- Create new script
- Rename script
- Create a copy of a script
- Edit script
- Execute script
- Customize diagnostic plots
- Delete script

Pirana includes functionality to run custom R-scripts on output from model executions. Scripts can be written by the user, but a considerable collection of scripts is also bundled with Pirana, which can serve as starting point for your own implementations.

The Scripts tab on the right side of the *Pirana* window is where you can find a list of available R scripts.



Basic_GOF
- Abs_IWRES_vs_IPRED.R
- CWRES_vs_PRED.R
- CWRES_vs_TIME.R
- Combined_GOF.R
- DV+IPRED_vs_TIME_individual.R
- DV+PRED_vs_TIME.R
- DV+PRED_vs_TIME_individual.R
- DV_vs_IPRED.R
- DV_v

Descr: This R-script create four basic GOF plots

Reqd: - NM table file with DV, PRED, IPRED, TIME, CWRES on first $TABLE record

Scripts are stored in three locations:

- **Library**: Group-wide scripts are stored in the `scripts` folder in the location where Pirana is installed.
- **My scripts**: User scripts are stored in `.pirana\scripts` in your home directory.
- **Project**: Project-specific scripts are stored in the subfolder `pirana_scripts` in the current folder. This subfolder is not created by default, it must be created by the user.

Use the pull-down in the tab's toolbar to select the location to list.

The folder structure underlying these scripts folders is reconstructed within the Scripts tab. A new folder can be added to the list by right-clicking in the Scripts tab and selecting the **New folder** menu option. Right-click on a selected folder to add a new subfolder.

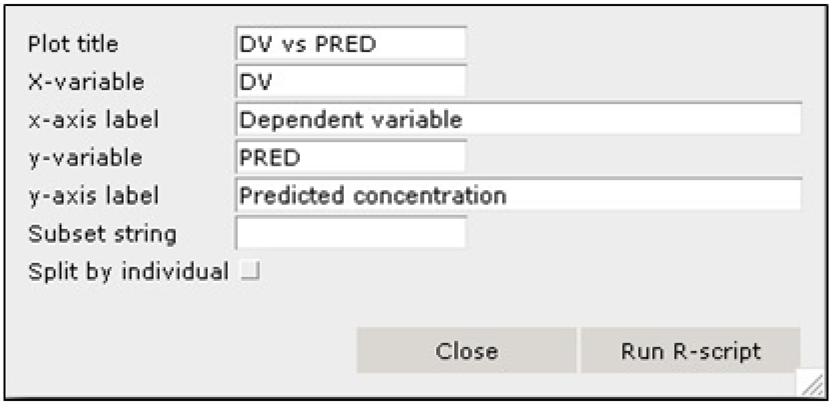
Click 🔄 in the Scripts tab toolbar to reload the script list.

Interactive scripts

Pirana has the ability to create ***interactive scripts***, meaning that upon execution of an R-script, the user will be presented with a dialog that asks for plotting and input options. The plotting options can be specified in the R-script. For example:

```
### <arguments>
###       <title label="Plot title">DV vs PRED</title>
###       <x_var label="X-variable">DV</x_var>
###       <xlab label="x-axis label">Dependent variable</xlab>
###       <y_var label="y-variable">PRED</y_var>
###       <ylab label="y-axis label">Pred. concentration</ylab>
###       <subset label="Subset string"></subset>
###       <split_id label="by ID" type="bool">FALSE</split_id>
### </arguments>
```

This will result in the following interface:



Plot title DV vs PRED
X-variable DV
x-axis label Dependent variable
y-variable PRED
y-axis label Predicted concentration
Subset string
Split by individual □

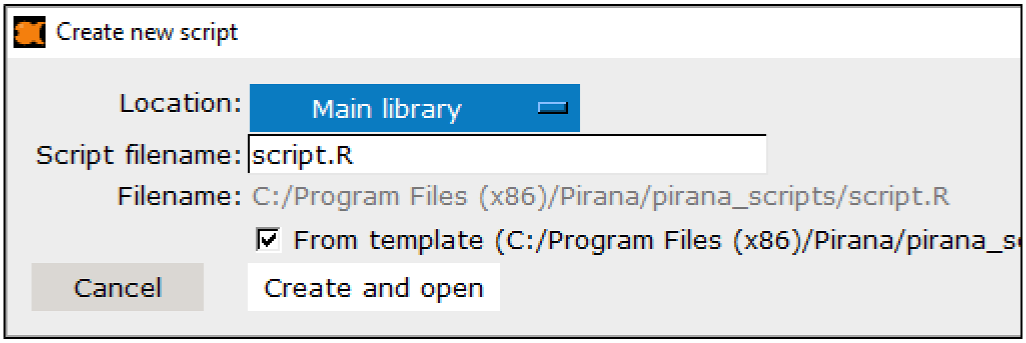
In the R-script, the specified options are then available as the list `arg` using:

```
ggplot (data=tab, aes (x=get(arg$x_var),
                       y=get(arg$y_var))) + geom_point()
```

Create new script

Pirana has an integrated library of R scripts (installed in `Pirana/pirana_scripts`, by default), which can be used to generate diagnostic plots based on model output files. The library of R scripts can also be easily edited or extended with new scripts. Thus, you can easily create a set of custom scripts that you can run against your specific analysis and tweak them to exactly what you want.

1. In the Scripts tab, click in the toolbar.
   Or
   Right-click in the Scripts tab and choose **New script** from the menu.



Location: Main library
Script filename: script.R
Filename: C:/Program Files (x86)/Pirana/pirana_scripts/script.R
☑ From template (C:/Program Files (x86)/Pirana/pirana_s

2. In the dialog, select the **Location** in which to store the new script.
3. Enter the filename to use for the new script in the **Script filename** field.
4. Check the **From template** box to load the template script in the text editor as a starting point. If unchecked, the text editor will be blank.
5. Press **Create and open** to open the text editor.

The template script has instructions that are commented into the script.

Rename script

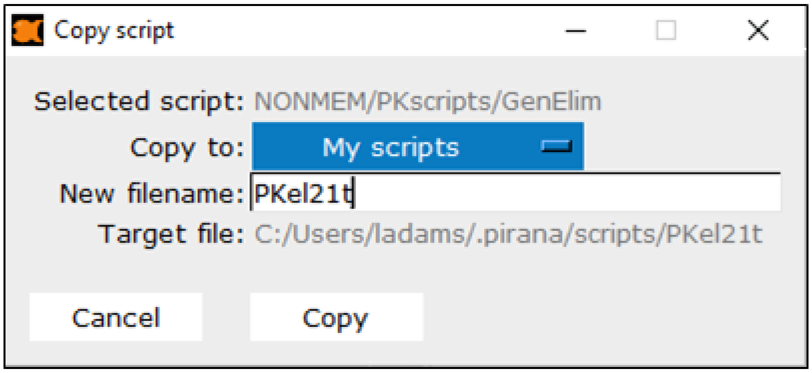
To change the name of a script:

1. In the Scripts tab, select the script to change.
2. Right-click the selected script and choose **Rename script** from the menu.
3. In the dialog, enter the new name in the **New script name** field.
4. Press **Rename**.

Create a copy of a script

To create a copy of a script and add it to one of the other two script locations:

1. In the Scripts tab, select the script to copy.
2. Right-click the selected script and choose **Copy script to** from the menu.



Selected script: NONMEM/PKscripts/GenElim
Copy to: My scripts
New filename: PKel21t
Target file: C:/Users/ladams/.pirana/scripts/PKel21t

3. In the dialog, select the location for the copied script from the **Copy to** pull-down.
4. In the **New filename** field, enter a name for the copied script.
5. Press **Copy**.

Edit script

Scripts can be edited either by doing one of the following:

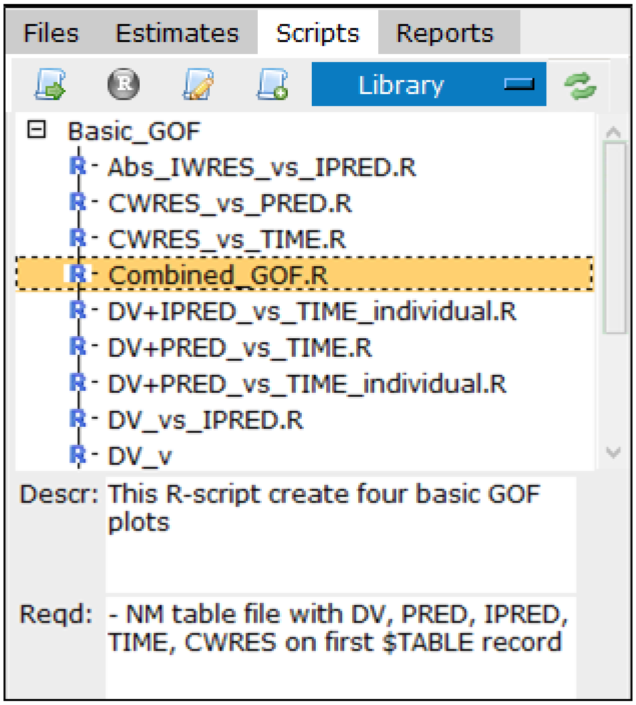
- – Edit them from outside Pirana.
- – In the Scripts tab, click ✎ in the toolbar.
- – Right-click the script in the Scripts tab and choose **Edit Script** from the menu.

Any changes made to the script will be saved permanently. You can choose to save the changes in the script that was originally supplied with Pirana, or save them to your own user library.

Note that when you install a new version of Pirana over the old one, any R script you have in the Pirana installation folder will be overwritten with the one supplied by the new version of Pirana (if you have not given the R-script another name).

Execute script

1. Select the model for which you want to create plots.
2. In the right panel, select the desired script from the Scripts tab.



Basic_GOF
- Abs_IWRES_vs_IPRED.R
- CWRES_vs_PRED.R
- CWRES_vs_TIME.R
- Combined_GOF.R
- DV+IPRED_vs_TIME_individual.R
- DV+PRED_vs_TIME.R
- DV+PRED_vs_TIME_individual.R
- DV_vs_IPRED.R
- DV_v

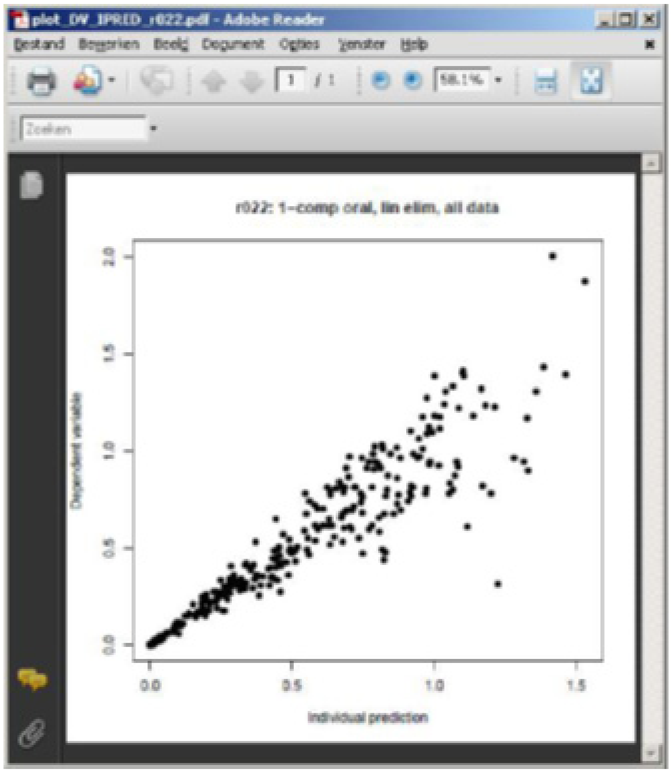
Descr: This R-script create four basic GOF plots

Reqd: - NM table file with DV, PRED, IPRED, TIME, CWRES on first $TABLE record

The expected input variables for the selected script are depicted at the bottom of the right panel.

3. Click .
   Or
   Right-click the selected script and choose **Run script** from the menu.

The plot will be created and opened automatically.


plot_DV_IPRED_r022.pdf - Adobe Reader
Bestand Bewerken Beeld Document Opties Venster Help
1 / 1
58.1%
Zoeken
r022: 1-comp oral, lin elim, all data
Dependent variable
0.0
0.5
1.0
1.5
2.0
Individual prediction
0.0
0.5
1.0
1.5

Pirana invokes R and runs the script in the directory `piranatemp` underlying the active folder. However, before execution, Pirana replaces `#PIRANA_IN` with an R list-object which specifies model and results information. For example:

```
models <- list (
   "003" = list (
      "modelfile"        = "003.mod",
      "description"      = "PK model digoxin",
      "reference_model" = "002",
      "data_file"        = "nm_pk_001.csv",
      "output_file"      = "003.lst",
      "tables"           = c("003.TAB", "sdtab003")
   )
)
```

Customize diagnostic plots

Instead of running the script through **Run Script** (discussed previously), the script may also be sent to the R Studio by doing one of the following:

- Click R in the Scripts tab toolbar.
- Right-click the selected script in the Scripts tab and choose **Open script in RStudio** from the menu.

The script will be opened in RStudio, where it may be further modified and then executed.

Delete script

Scripts that you have created can be deleted as follows:

1. In the Scripts tab, select the script to delete.
2. Right-click the selected script and choose **Delete script** from the menu.

Reporting Functionality

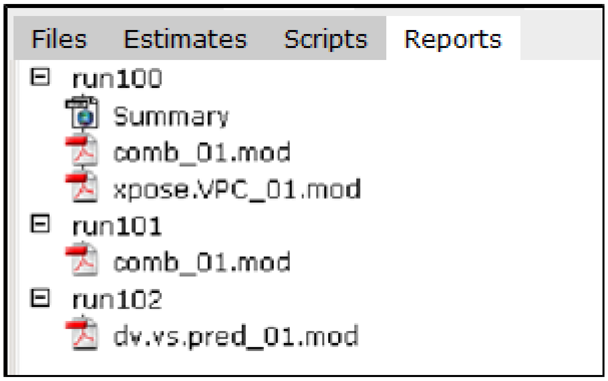
The following functionality for reporting is available in Pirana:

- Run reports in various formats (HTML, Word, text, LaTeX, pdfLaTeX)
- Run records of single and multiple runs (CSV/Excel, HTML, Word, text)
- Visual Run Record
- Execution log files of executed runs (see View execution log)

Run reports

Run reports with model and run information as well as parameter estimates can automatically be generated and outputted as HTML, LaTeX, Word, or plain text format. The report optionally displays basic run information, run statistics, description, notes, and parameter estimations, split by implemented estimation methods. The information to include in the report can be specified in **File > Settings > Run reports** (see “Run reports settings” for details).

After a run report is generated, it will show up in the list on the right, under the **Reports** tab. In this tab, goodness of fit plots, generated either using the Xpose GUI in Pirana or the R scripts library, are also shown. Double-clicking on any of the plots or reports will re-open them.



⊟ run100
Summary
comb_01.mod
xpose.VPC_01.mod
⊟ run101
comb_01.mod
⊟ run102
dv.vs.pred_01.mod

Note: In the run reports, Pirana calculates the RSE for population parameters as

$$RSE_{\theta_i} = \frac{SD_{cov,\theta}}{\theta_i}$$

but does not take into account log-transformation of parameters (e.g., when MU-modeling). For inter-individual and residual variance (Ω and Σ), RSEs are calculated as

$$RSE_{\omega^2_{i,i}} = \frac{SD_{cov,\omega^2_{i,i}}}{\omega^2_{i,i}}$$

RSEs given for $\omega_{i,i}$ and $\sigma_{i,i}$ are calculated as

$$RSE_{\omega_{i,i}} = \frac{SD_{cov,\omega^2_{i,i}}}{2 \cdot \omega^2_{i,i}}$$

1. In the *Pirana* window, select the executed model in the list.

2. Click in the main toolbar.
   Or

   Click in the Estimates tab toolbar.
   Or
   Right-click the selected model and choose **Reports** in the menu.
   Or
   Select the **Results > Run Reports** menu option.

3. Choose the format from the submenu.

   HTML

   Microsoft Word

   Plain text

   LaTeX

   pdfLaTeX

The report will open up in the related software. (LaTeX output is opened in the specified code editor, but also can be converted automatically to PDF using **pdflatex**, if installed.) Generated run reports are placed into a Pirana reports folder.

See the “Reports” automated workflow example for another way to generate a Word report (**Tools > Automated modeling workflow > Report** menu option).

Run records

For all runs in a project folder, or a subset thereof, a CSV run record can be compiled by Pirana that includes all model and run characteristics, such as model description, estimation method, objective function value, termination result, etc. An abridged version of the run record can also be created as a plain text or Word document.

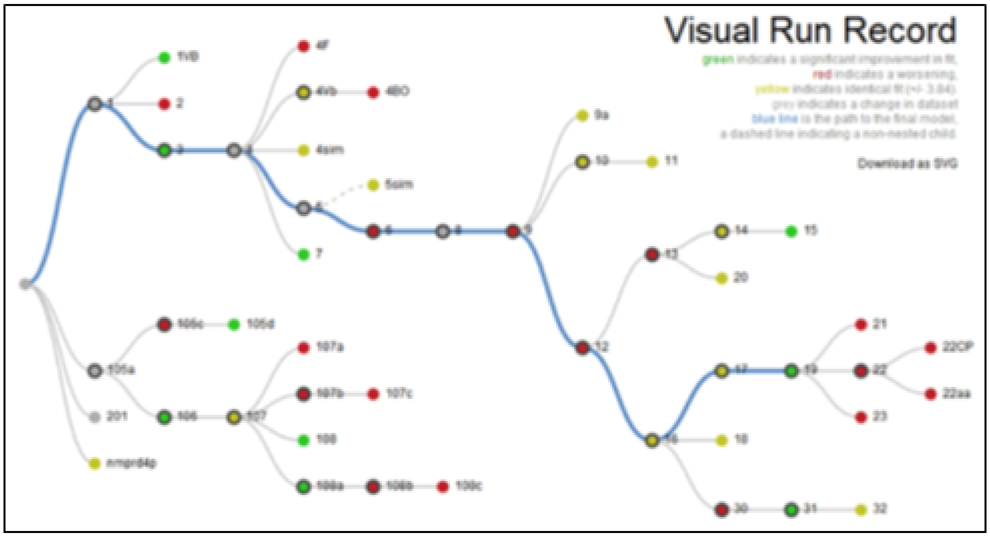
1. Select the project folder or the runs to include in the output CSV file.
2. Select the format from the **Results > Run records** menu.
   - **Visual Run Record**
   - **HTML Run Record**
   - **Detailed Run Record (csv)**
   - **Brief Run Record (Word)**
   - **Brief Run Record (text file)**
   - **Run Record (PsN)** - available for NONMEM models only

Visual Run Record

Pharmacometric model development most often progresses in a hierarchical fashion, using the likelihood ratio test to assess significance of improved fit between nested models. An appropriate visualization of the model hierarchy can help in gaining a better understanding of key stages in model building, and can aid in communicating the model development history to others.

Pirana can generate a visual run record (VRR) for visualizing the model building process from initial to final model. The VRR is an SVG file that contains in interactive tree view of the model development process. Models are related to each other based on Pirana's reference model tags in the model file.

1. Select the **Results** > **Run records** > **Visual run record** menu option.
2. Select a final model.


Visual Run Record
green indicates a significant improvement in fit,
red indicates a worsening,
yellow indicates identical fit (+/- 3.84),
grey indicates a change in dataset
blue line is the path to the final model,
a dashed line indicating a non-nested child
Download as SVG
1VB
2
4f
4Vb
4BO
4sim
5sim
7
9a
10
11
14
15
13
20
12
21
22CP
22aa
17
19
22
23
16
18
30
31
32
105c
105d
105a
201
106
107
107a
107b
107c
108
108a
108b
108c
nmprd4p

Colors aid in visualizing the improvement/worsening of model fit (green/red), and whether the model has children or not. In each branch, the nodes are ordered by OFV. When a final model is specified, the modeling path can be made visible as a blue line, thereby easily identifying the key runs.

NLME Models

- Create or edit a model using Model Builder
- Generate a new model using a template
- Run initial estimates
- Duplicate models
- Batch editing of models
  (refer to the Pirana “Batch editing of models” section)
- Open input dataset
  (refer to the Pirana “Open the input dataset” section)
- Attach folders to a model
  (refer to the Pirana “Attach folders to selected model” section)
- Rename a model file
- View difference with reference model
  (refer to the Pirana “View difference with reference model” section)
- Delete a model
  (refer to the Pirana “Delete a model” section)
- Fit a model
- Execute a model run with Model Executor
- View model results

Create or edit a model using Model Builder

The RsNLME *Model Builder* is a Shiny app that breaks up the NLME model building process into several steps. As you make selections, the interface will update automatically, adding and removing options that become valid or invalid, respectively, based on your choices.

To create a new NLME model:

1. Select **Models > New model**.
   Or
   Select and right-click an existing NLME model in the main *Pirana* window and choose **RsNLME shiny > New Metamodel** from the menu.
2. Press the **Model Builder** button.
3. Specify the input source by entering the full path to the .csv or .txt file or click 📁 to use the file explorer.
4. Click **Run model builder**.

The RsNLME *Model Builder* opens in a browser window.

Refer to the RsNLME user documentation for more details about the *Model Builder* app.

To edit an NLME model:

1. Select the model to edit in the main *Pirana* window, the choose **Models > Edit model**.
   Or

Select and right-click an existing NLME model and choose **RsNLME shiny > Edit Metamodel** from the menu.

The RsNLME *Textual Model Builder* opens in a browser window. Edit the model code directly in the editor (click **SAVE** when finished). You can also use the tabs on the right to adjust the column mapping, input options, or input data.

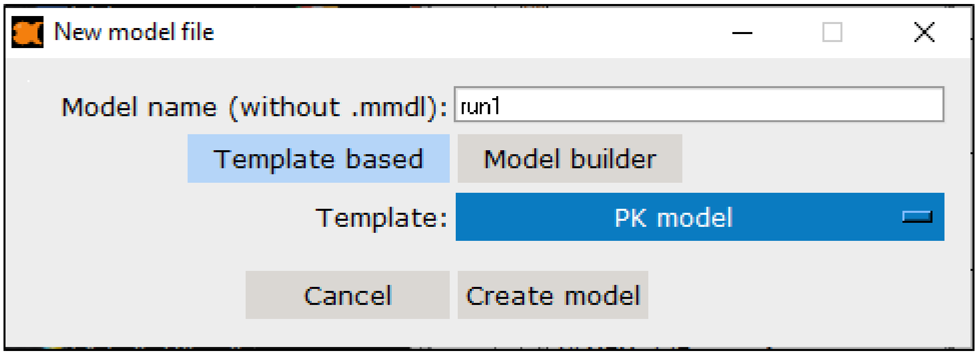
Refer to the RsNLME user documentation for more details about the *Textual Model Builder* app.

Generate a new model using a template

Many basic template models are included in Pirana. It is also possible to build your own library with base models that you often use. Templates can be added by copying a model file to `/templates` in the `Pirana` directory. The template models should have the same file extension as your model files to be recognized as a template.

1. Select **Models > New model**.
2. Press the **Template based** button
3. Select the model type from the **Template** pulldown and click **Create model**.

The new model will open in the text editor defined in the Software integration Configuration Settings page.



Model name (without .mmdl): run1

Template: PK model

Run initial estimates

1. Select the model in the list.
2. Right-click the selected model and choose **RsNLME shiny > Run InitialEstimates** from the menu.

The *InitialEstimates* Shiny app opens in a browser window. The current initial estimates are loaded and a graph generated. The graph is instantaneously updated as adjustments to settings are made using the provided tools.

Refer to the RsNLME user documentation for more details about the *InitialEstimates* app.

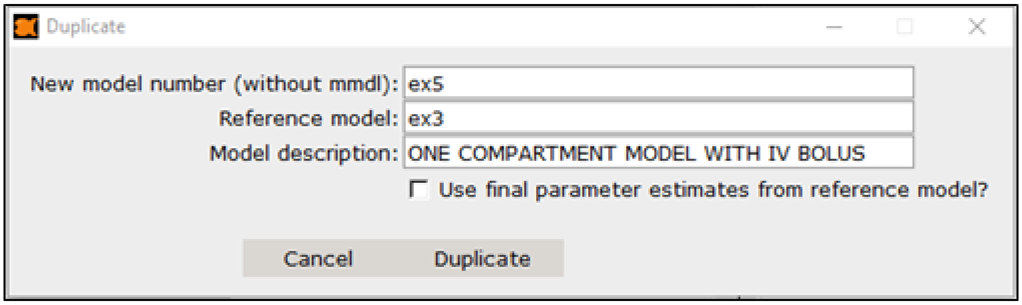
Duplicate models

- Create a single duplicate of a model
- Create multiple duplicates of multiple models

Create a single duplicate of a model

1. Select the model in the list.
2. Right-click the selected model and choose **File actions > Duplicate** from the menu.
   Or
   Select the **Models > Duplicate model** menu option.
   Or

   Click in the toolbar.



New model number (without mmdl): ex5
Reference model: ex3
Model description: ONE COMPARTMENT MODEL WITH IV BOLUS
□ Use final parameter estimates from reference model?

3. In the *Duplicate* window, enter the model number in the **New model number** field.
4. Change number of the **Reference model** for the new model, if desired.
   By default, this will take the name of the original model selected for duplicating.
5. Adjust the **Model description** for the new model.
6. Check the **Use final parameter estimates from reference model** box to update parameter estimates in the new model to the ones estimated for the original model.

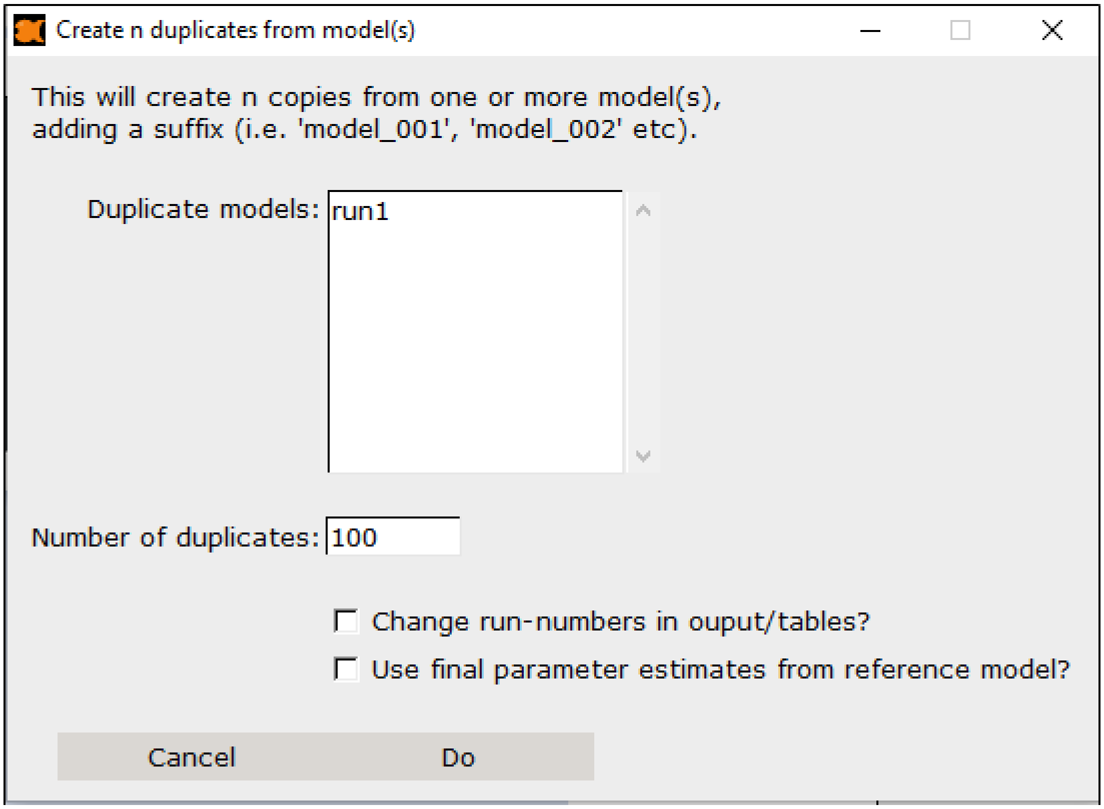
After pressing the **Duplicate** button, a new model is created and opened in the editor.

Create multiple duplicates of multiple models

Select the **Tools > Batch operations > Create n duplicates of model(s)** menu option.
Or

Click in the toolbar and select **Create n duplicates of model(s)** from the menu.

Creates multiple duplicates of multiple model files, with (optionally) updated run/table numbers and final parameter estimates.



Create n duplicates from model(s)

This will create n copies from one or more model(s),
adding a suffix (i.e. 'model_001', 'model_002' etc).

Duplicate models: run1

Number of duplicates: 100

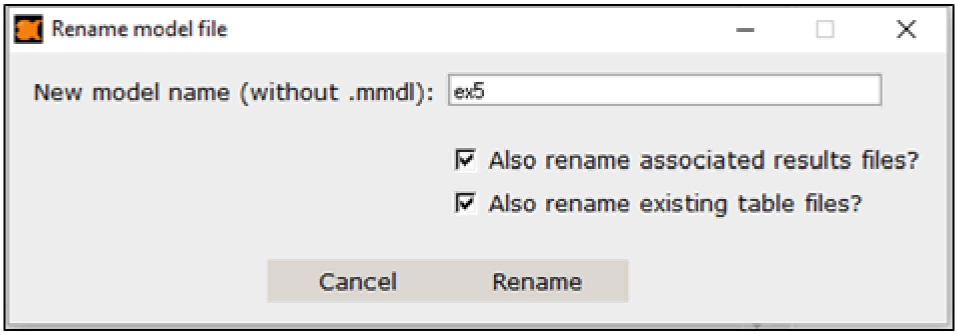
□ Change run-numbers in ouput/tables?

□ Use final parameter estimates from reference model?

1. Select the model(s) to be duplicated from the **Duplicate models** list.
2. Enter the **Number of duplicates** for each selected model in the field.
3. Check the **Change run-numbers in output/tables** box to incrementally update the run numbers for each duplicate.
4. Check the **Use final parameter estimates from reference model** box to update parameter estimates in the new models to the ones estimated for the original model.

Rename a model file

1. Select the model in the list.
2. Right-click the selected model and choose **File actions > Rename** from the menu.
   Or
   Select the **Models > Rename model** menu option.
   Or
   Click ✏ in the toolbar.



Rename model file

New model name (without .mmdl): ex5

☑ Also rename associated results files?

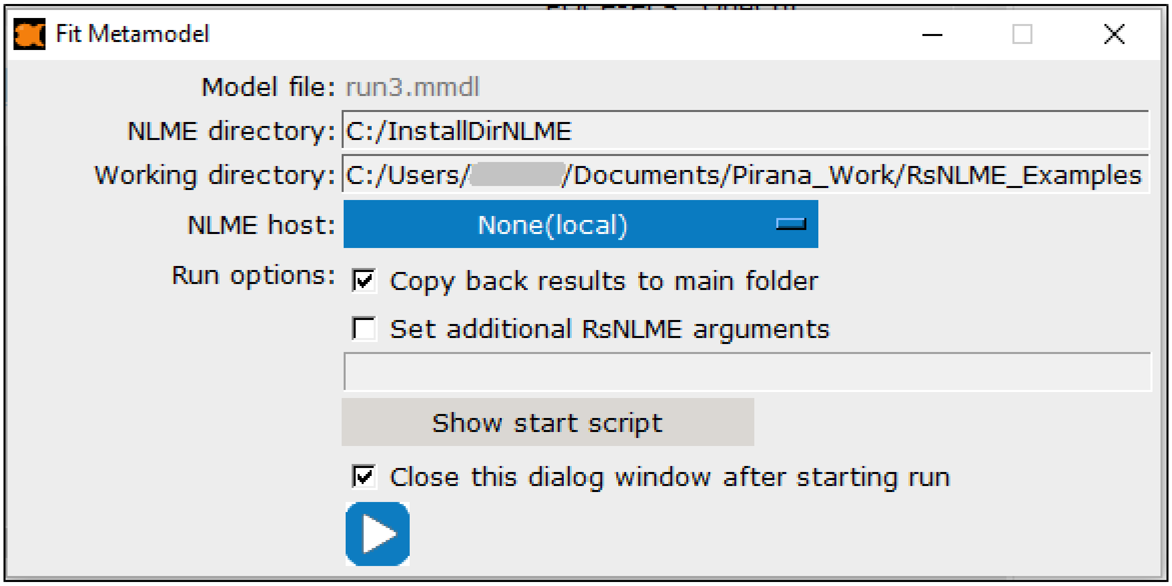
☑ Also rename existing table files?

3. Edit the text in the **New model name** field.
4. To maintain the connection with the model using the new name, check the boxes to rename any results and/or table files associated with the model.

Fit a model

Executes a model using a predefined start script.

1. Select the NLME model to run in the main *Pirana* window.
2. Right-click the selected model and choose **Fit Metamodel**.



Fit Metamodel

Model file: run3.mmdl

NLME directory: C:/InstallDirNLME

Working directory: C:/Users/[redacted]/Documents/Pirana_Work/RsNLME_Examples

NLME host: None(local)

Run options: ☑ Copy back results to main folder

□ Set additional RsNLME arguments

☑ Close this dialog window after starting run

The dialog shows the name of the model file, the location of the NLME executables, and the current working directory in which the model file is located.

3. Select the **NLME Host** where the model fit is to be executed from the list of pre-specified profiles.

   The contents of this menu can be modified using preference settings, refer to the “NLME Settings” section in the Pirana user documentation.

4. Check the **Copy back results to main folder** box to put results files in the current working directory.
5. Check the **Set additional RsNLME arguments** box to provide additional arguments to the R script command to start R in the field below the checkbox.
6. Click the **Show start script**/**Hide start script** button to toggle display of the following:
   - Name of the batch file containing the start script
   - The command (with additional arguments) for the batch file
   - The editable text of the batch script to run
7. Check the **Close this dialog window after starting run** box to automatically close the window once ▶ is clicked.

Execution is done in a separate folder that is attached to the model.

Execute a model run with Model Executor

The *Model Executor* Shiny app is used to review and adjust run options and then execute the run.

1. Select the NLME model to run in the main *Pirana* window.
2. Right-click the selected model and choose **RsNLME shiny > Execute Metamodel**.

The RsNLME *Model Executor* opens in a browser window.

Refer to the RsNLME user documentation for more details about the *Model Executor* app.

View model results

1. In the *Pirana* window, double-click the NLME Model folder in the list.

   The output files from the model run are listed in the Files tab on the right side of the window. They include .csv and .txt files. Descriptions of the .csv results files can be found in the Phoenix NLME user documentation.

2. Double-click a file to view the contents.

There are numerous plot scripts available in Pirana that you can use to graph the results. Refer to “R scripts for graphs and file processing” for more information.

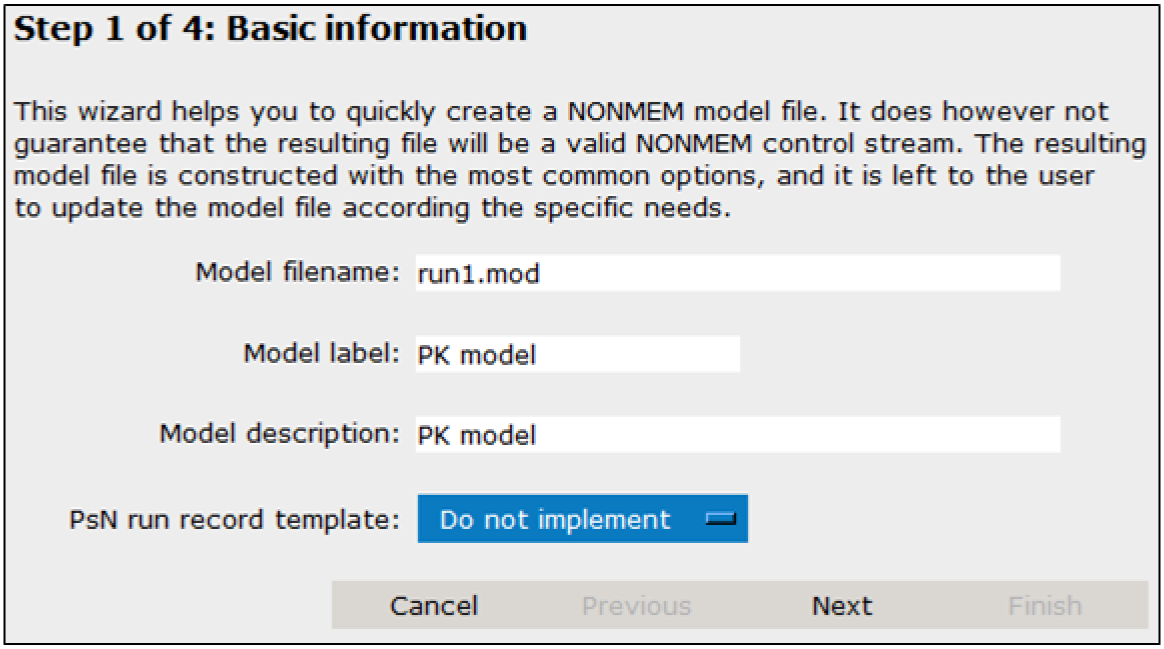
NONMEM Models

Create NONMEM models

Create a NONMEM PK model file using a wizard

The PK Model wizard allows stepwise construction of a range of PK models in NONMEM. It includes the most commonly used ADVANs, estimation methods, and the most commonly used residual error models. Of course, keep in mind that you have to change the initial estimates and the $DATA and $INPUT records to suit your PK problem.

1. Select **Tools > Wizards**.
2. In the dialog, choose **PK model (NONMEM)** and click **Run Wizard**.



Step 1 of 4: Basic information

This wizard helps you to quickly create a NONMEM model file. It does however not guarantee that the resulting file will be a valid NONMEM control stream. The resulting model file is constructed with the most common options, and it is left to the user to update the model file according the specific needs.

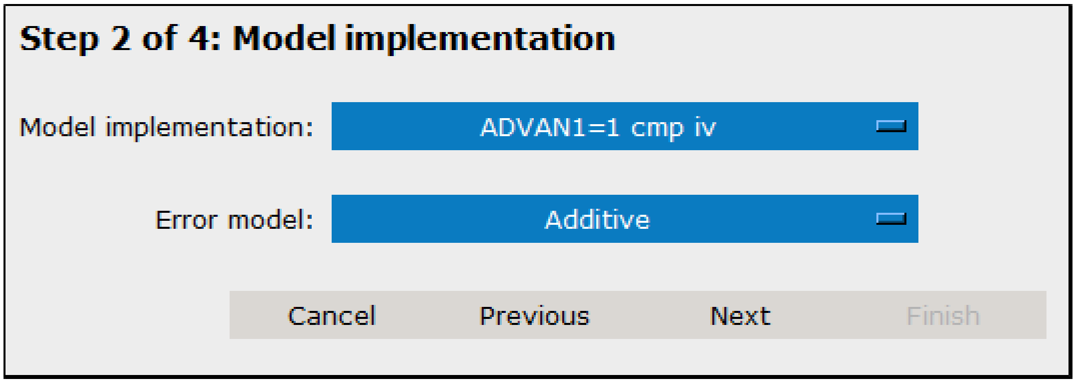
Model filename: run1.mod

Model label: PK model

Model description: PK model

PsN run record template: Do not implement

3. Enter the name for the file in the **Model filename** field.
4. Enter a label to use when referring to the model in the **Model label** field.
5. Use the **Model description** field to enter a description for the model.
6. Use the **PsN run record template** pull-down to indicate if run records are to be created using PsN's `runrecord` command, which allows customizable run record tables to be generated. See PsN documentation for additional information.
7. Press **Next**.



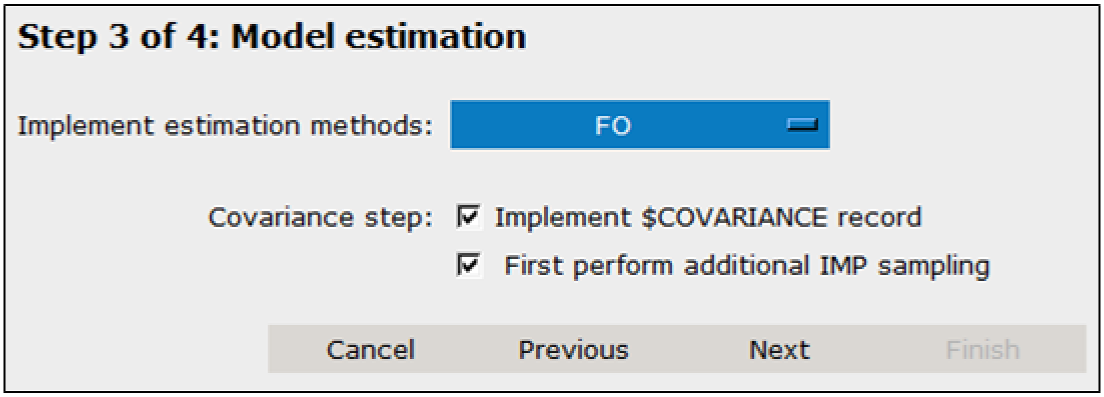
Step 2 of 4: Model implementation

Model implementation: ADVAN1=1 cmp iv

Error model: Additive

8. From the **Model implementation** pull-down, select the desired NONMEM model.

9. From the **Error model** pull-down, choose the type of error model to apply: additive, proportional, combined additive and proportional, exponential, or log-transform both sides.
10. Press **Next**.



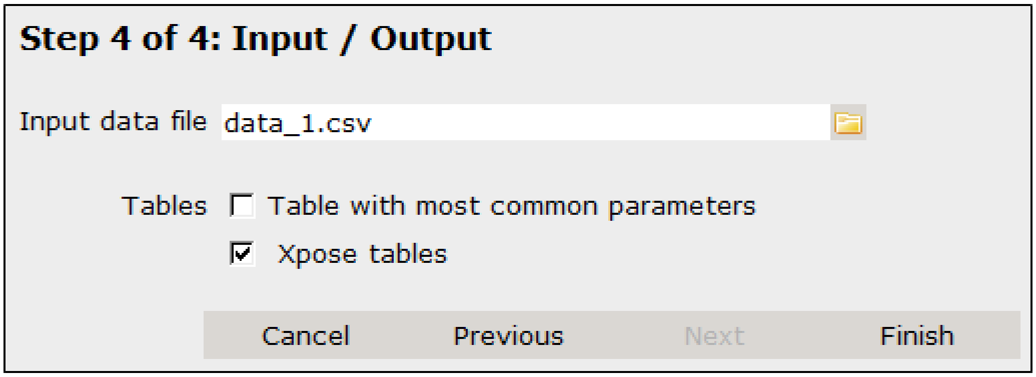
Step 3 of 4: Model estimation

Implement estimation methods: FO

Covariance step: ☑ Implement $COVARIANCE record

☑ First perform additional IMP sampling

11. From the **Implement estimation methods** pull-down, select the estimation method to use.
12. Check the **Implement $COVARIANCE record** to obtain standard errors of the parameter estimates.
13. Check the **First perform additional IMP sampling** to perform initial importance-sampling prior to estimation.
14. Press **Next**.



Step 4 of 4: Input / Output

Input data file data_1.csv

Tables □ Table with most common parameters

☑ Xpose tables

15. Specify the Input data file by either typing in the field or click 📁 to use the file browser.
16. Check the **Table with most common parameters** box to generate this table as part of the output.
17. Check the **Xpose tables** box to include Xpose tables as part of the output.
18. Press **Finish**.

The model file is saved to the current folder.

Generate a new model using a template

Many basic template models are included in Pirana. It is also possible to build your own library with base models that you often use. Templates can be added by copying a model file to `/templates` in the `Pirana` directory. The template models should have the same file extension as your model files to be recognized as a template.

1. Select **Models > New model**.
2. Choose the **Template** model to use and click **Create model**.

The new model will open in the text editor defined in the Software integration Configuration Settings page.

NONMEM template control file syntax

There are some important syntax rules to following in the NONMEM template control file that make Pirana/PsN/Xpose work best together. These involve tags, parameters, and naming rules for output tables.

- Users of PsN and Xpose likely follow the 'Uppsala convention' of having model files named like `run1.mod`, `run2.mod`, etc. This is recommended for Pirana users as well, although Pirana is flexible in this respect. Note that Pirana removes the run from the model file name in the model overview.
- Use the default file extensions to ensure that the programs recognize the necessary files. For example: .mod for NONMEM model files, .tab for NONMEM $TABLE data, .mmdl for NLME meta-model files, .mdl for NLME model files.
- Pirana looks for a description of the model in the first part of the control file. It adheres to PsN's run record standards. If the PsN run record is not used, Pirana searches for the words `$PROBLEM` or `Model desc:` to extract the model description.
- If you want to use the hierarchy functionality for models, you should specify the reference model in the first few lines of the control file. Again, it is best to use PsN's run record specification, but Pirana is flexible and also compatible with Census, and understands the following syntaxes:

```
;; 1. Based on: 001.mod
; Ref. model:001.mod
; Ref:001.mod
; Parent=001.mod
```

- Model parameter descriptions need to be specified after a semi-colon, e.g.

```
$THETA
(3, 5, 11) ; CL/F
(10, 50, 100) ; V/F
```

  Note that Pirana reads these descriptions from the control file (and not from the output file). To be read correctly, covariance block need to be specified as:

```
$OMEGA BLOCK(2) 0.1 ; IIV CL/F
0.05 ; COV CL~V 0.1 ; IIV V/F
```

  or as:

```
$OMEGA BLOCK(3)
0.1           ; IIV CL/F
```

```
0.05 0.1      ; IIV V/F
0.01 0.05 0.1 ; IIV KA
```

- When models are to be executed in a separate directory, files needed for compilation (e.g., additional Fortran routines in `.FOR` files), are copied automatically by Pirana. These files should be specified in the `OTHER` and `CONTR` entries on the `$DATA` record. If additional files are needed, you can instruct Pirana to copy these by adding this line to your control stream:

  ```
  ; INCLUDE=file1_to_be_copied.ext,file2_to_be_copied.ext, ...
  ```

  Note that PsN has its own functionality for doing this.

- In general, Xpose needs four table files to be specified to support most of its diagnostic plotting capabilities:

  sdtab: Standard output table that includes all time records, e.g.:
  ```
  $TABLE ID TIME IPRED IWRES CWRES EVID MDV NPD NPDE NOPRINT
       ONEHEADER FILE=sdtab1
  ```
  (Note the “1” affixed to the `FILE=sdtab1` — this corresponds to the run number for the file.)

  patab: Parameter table for model parameters, one record per parameter set per individual, e.g.:
  ```
  $TABLE ID CL V KA TVCL TVV TVKA ETA1 ETA2 ETA3 NOPRINT ONEHEADER
       FILE=patab1
  ```

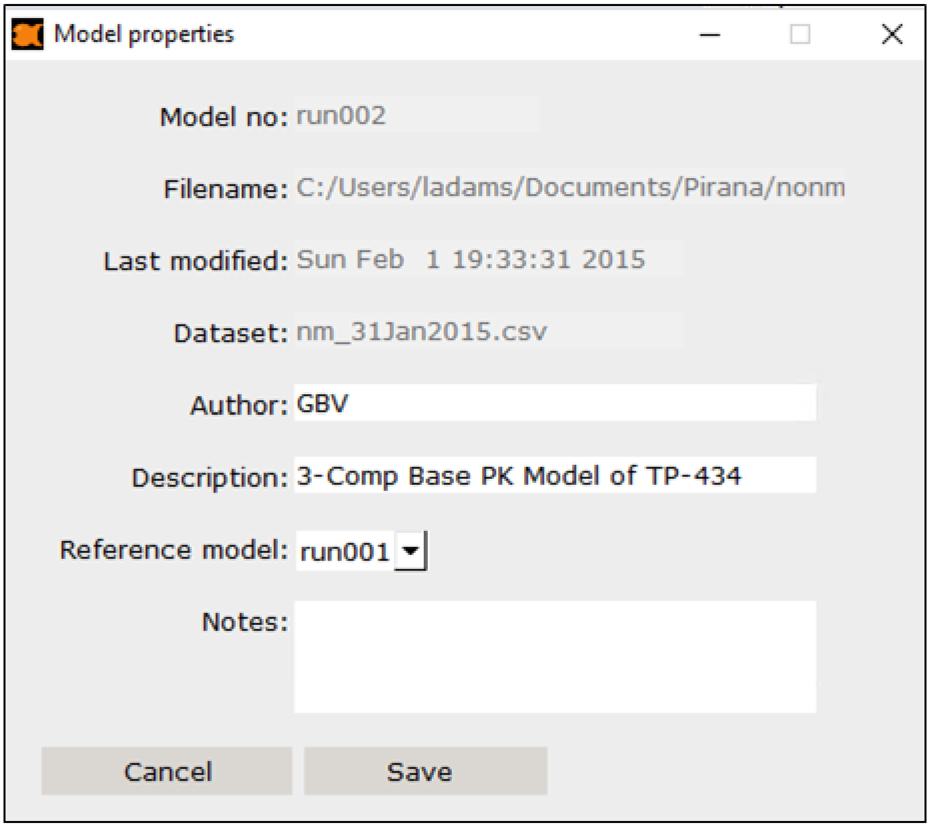
  cotab: Continuous covariate table for continuous covariate variables, e.g.:
  ```
  $TABLE ID WT AGE CLCR NOPRINT ONEHEADER FILE=cotab1
  ```

  catab: Categorical covariate table for categorical covariates, e.g.:
  ```
  $TABLE ID SEX RACE FOOD NOPRINT ONEHEADER FILE=catab1
  ```

- NONMEM template control files, when planning to use the **Results > Run records > Run record (PsN)** function, are considered to be full templates, as they contain the model and all of the additional information that will become part of the run record output. The short template has the model information only.

Add notes and view model properties

1. Select the model in the list.
2. Right-click the selected model and choose **Model > Notes and Info** from the menu.
   Or
   Select **Models > Model properties** from the main menu.
   Or
   Select ⓘ in the toolbar.
   Or
   Use the **Ctrl-I** shortcut.



Model no: run002

Filename: C:/Users/ladams/Documents/Pirana/nonm

Last modified: Sun Feb 1 19:33:31 2015

Dataset: nm_31Jan2015.csv

Author: GBV

Description: 3-Comp Base PK Model of TP-434

Reference model: run001

Notes:

Basic information about the selected model is displayed in the dialog. The model number, filename, last modified date, and associated dataset are read-only.

3. Change the name of the model's author in the **Author** field.
4. Enter or edit the model's description in the **Description** field.
5. Select a **Reference** model for the current model from the pull-down.
6. Enter any additional information about the model in the **Notes** field.

Notes can be removed by either deleting the text in the *Model Properties* dialog or selecting **Model > Remove notes** from the model's right-click menu.

Note: The notes are stored in a database file (`pirana.dir`), which is created automatically in each folder that holds models and is visited by Pirana. If you archive your projects manually, make sure to include these files as well.

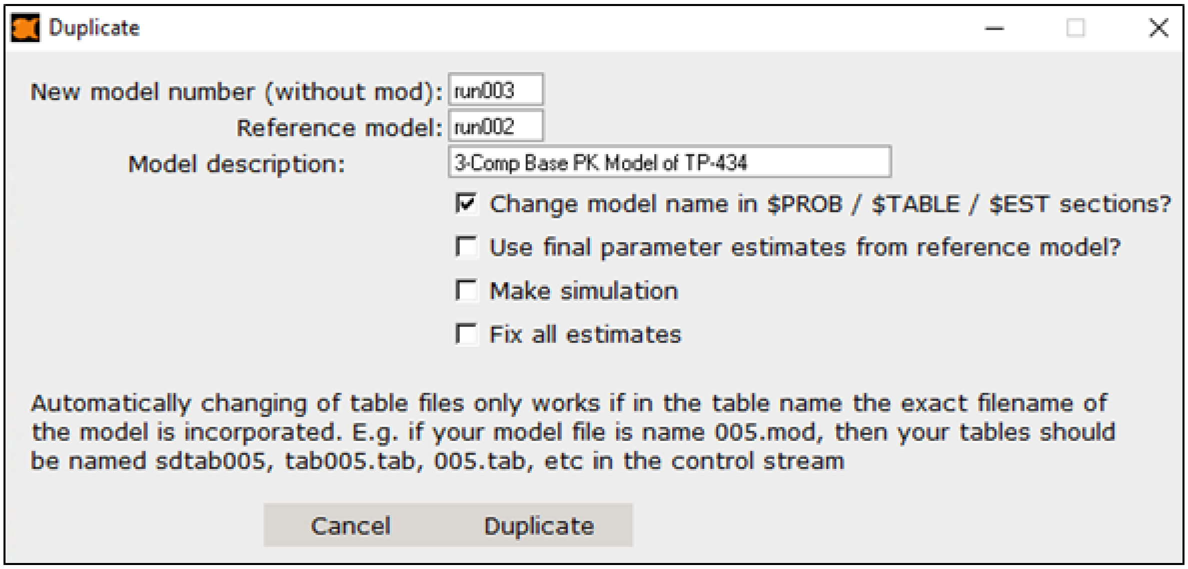
Duplicate models

- Create a single duplicate of a model
- Create multiple duplicates of multiple models
- Create a duplicate for MSF restart

Create a single duplicate of a model

1. Select the model in the list.
2. Right-click the selected model and choose **File actions > Duplicate** from the menu.
   Or
   Select the **Models > Duplicate model** menu option.
   Or

   Click in the toolbar.



New model number (without mod): run003
Reference model: run002
Model description: 3-Comp Base PK Model of TP-434

☑ Change model name in $PROB / $TABLE / $EST sections?
☐ Use final parameter estimates from reference model?
☐ Make simulation
☐ Fix all estimates

Automatically changing of table files only works if in the table name the exact filename of the model is incorporated. E.g. if your model file is name 005.mod, then your tables should be named sdtab005, tab005.tab, 005.tab, etc in the control stream

3. In the *Duplicate* window, enter the model number in the **New model number** field.
4. Change number of the **Reference model** for the new model, if desired.
   By default, this will take the name of the original model selected for duplicating.
5. Adjust the **Model description** for the new model.
6. Check the **Change model name in...** box to update the model name within sections of the model file itself with the new model number
7. Check the **Use final parameter estimates from reference model** box to update parameter estimates in the new model to the ones estimated for the original model.
8. Check the **Make simulation** box to make the copied model a simulation model.
   When checked, the estimation and covariance steps ($EST and $COV, if present) records are commented out with a semicolon, and a simulation record is inserted into the control file, e.g.:
   `$SIM (12345) (54321) ONLY SIM`.
9. Check the **Fix all estimates** box to use the same estimates in the copied model as in the original.
   When checked, $THETA, $OMEGA, and $SIGMA records are modified with the addition of the text “FIX” to fix the parameter estimates, e.g.:
   `$THETA`

```
(0, 1.5,4) FIX ;KA
(0.1, 0.377,5) FIX ;V/F
(0, 0.0327,1) FIX ;CL/F
(-2, 1.98,2) FIX ;WT on KA

$OMEGA
0.281 FIX ; IIV KA
0.0274 FIX ; IIV V/F
0.0516 FIX ; IIV CL/F

$SIGMA
0.0256 FIX ; properr
```

Please note that, to correctly duplicate with updated parameter estimates, you are required to adhere to some coding guidelines, especially for the $OMEGA and $SIGMA blocks. See the "NONMEM template control file syntax" for more information.

After pressing the **Duplicate** button, a new model is created and opened in the editor.

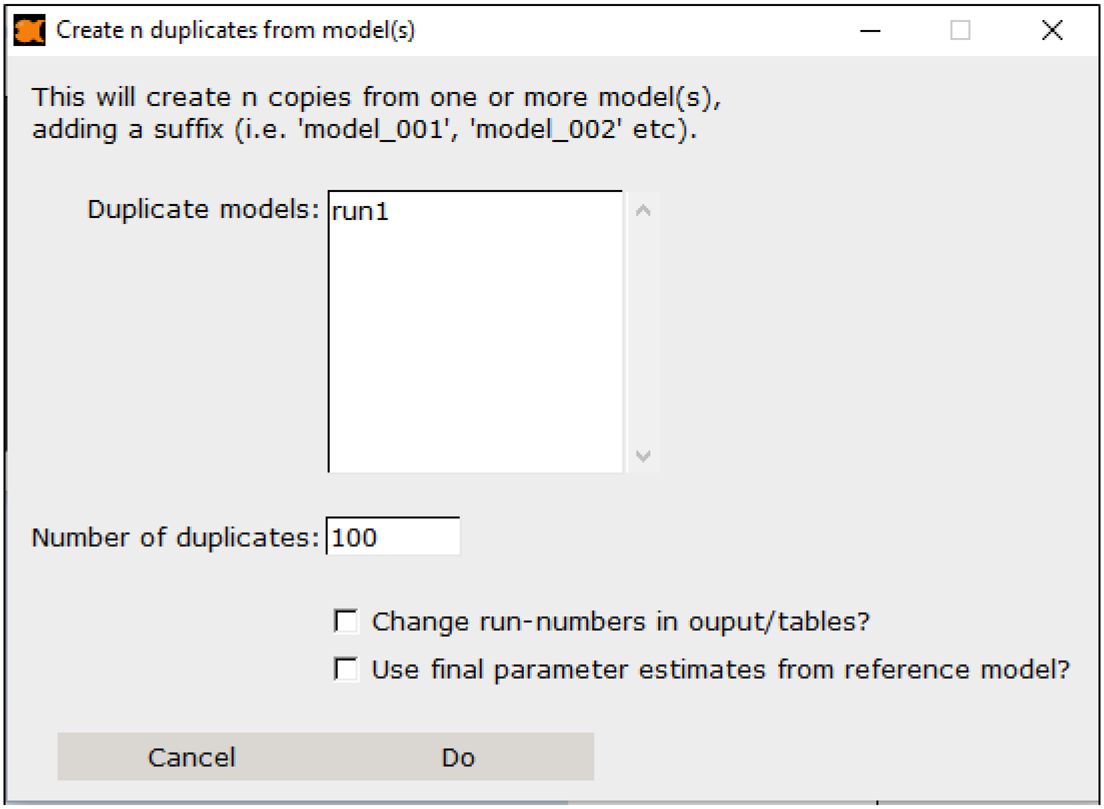
Note: A useful output file generated when MSF option is used is called INTER, which is an easy read format of the output from NONMEM. Iteration by iteration is listed so the analyst can get ideas for where the run is struggling.

Create multiple duplicates of multiple models

Select the **Tools > Batch operations > Create n duplicates of model(s)** menu option.
Or

Click in the toolbar and select **Create n duplicates of model(s)** from the menu.

Creates multiple duplicates of multiple model files, with (optionally) updated run/table numbers and final parameter estimates.



Create n duplicates from model(s)

This will create n copies from one or more model(s), adding a suffix (i.e. 'model_001', 'model_002' etc).

Duplicate models: run1

Number of duplicates: 100

□ Change run-numbers in ouput/tables?

□ Use final parameter estimates from reference model?

1. Select the model(s) to be duplicated from the **Duplicate models** list.
2. Enter the **Number of duplicates** for each selected model in the field.
3. Check the **Change run-numbers in output/tables** box to incrementally update the run numbers for each duplicate.
4. Check the **Use final parameter estimates from reference model** box to update parameter estimates in the new models to the ones estimated for the original model.

Create a duplicate for MSF restart

A model can be duplicated for MSF restart**.** This means that the model file is duplicated, but an $MSFI record is added, parameter estimate blocks are commented out, and the $MSFO record is updated.

Restarting a NONMEM run using a MSF file requires that the original model included a MSF command in the estimation record, e.g.:

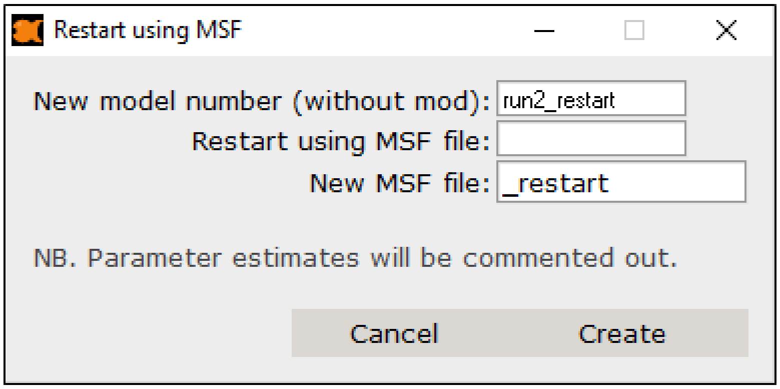
```
$EST MAXEVAL=9999 PRINT=5 MSF=run4.msf METHOD=CONDITIONAL INTER
```

When the MSF command is included in an estimation run, NONMEM will produce some additional files, two of which (`run4.msf` and `run4_ETAS.msf`) are used to restart a NONMEM run from where the previous run terminated using the control file created with this feature.

Note that the files `run4.msf` and `run4_ETAS.msf` must be copied into the root project folder directory from the previous run's subdirectory, where complete output run records are stored, e.g., `modelfit_dir4/NM_run1/run4.msf`. Although `run4_ETAs.msf` is not required, NONMEM will throw a warning if it is not in the root project directory.

1. Select the model in the list.

2. Right-click the selected model and choose **File actions > Duplicate for MSF restart** from the menu.
   Or
   Select the **Models > Duplicate model for MSF restart** menu option.
   Or
   Click in the toolbar.



New model number (without mod): run2_restart
Restart using MSF file:
New MSF file: _restart

NB. Parameter estimates will be commented out.

3. In the *Restart using MSF* window, enter the model number for the duplicate model in the **New model number** field.
4. Enter the name of the MSF file specified in the previous run's $EST record in the **Restart using MSF file** field.
5. Enter the filename for the **New MSF file** to create during the restart in the field.

Batch editing of models

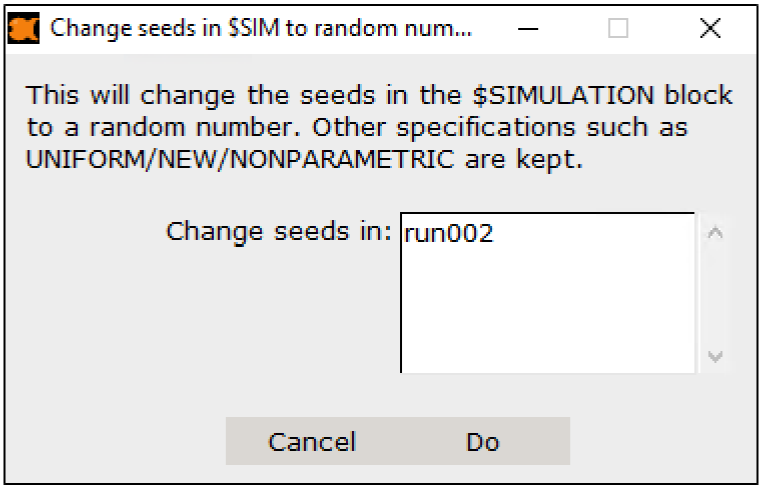
In addition to the following batch editing options, which are discussed in the Pirana user documentation, Pirana also allow batch editing of Random simulation seeds.

- Search and replace in models
- Change dataset
- Replace block in multiple models
- Add code to multiple models
- Add code to blocks in multiple models

Random simulation seeds

Select the **Tools > Batch operations > Random seeds in $SIM** menu option.
Or

Click in the toolbar and select **Random seeds in $SIM** from the menu.



This will change the seeds in the $SIMULATION block to a random number. Other specifications such as UNIFORM/NEW/NONPARAMETRIC are kept.

Change seeds in: run002

In all selected models, the $SIMULATION block will be updated with new seeds.

Datasets

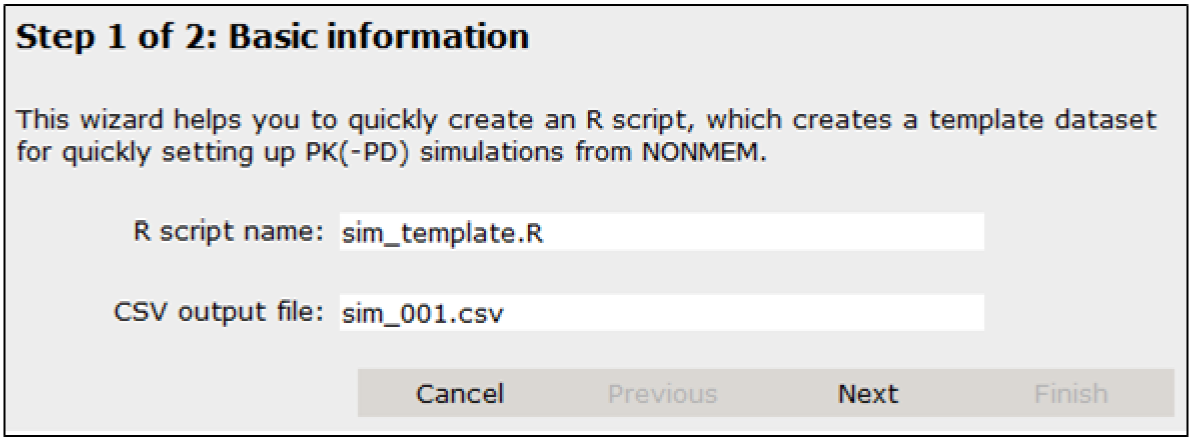
In addition to the “Create dataset template R script” wizard discussed below, Pirana also allows users to:

- Open the input dataset
- Attach folders to selected model

Create dataset template R script

This Wizard can be used to create an R script that, in turn, generates a NONMEM simulation data file with specified number of individuals, doses, observations, dosing times, and covariates. This can be useful for quickly setting up simulations in NONMEM.

1. Select the **Models > Wizards** menu option.
2. Choose **Dataset template** and click **OK**.



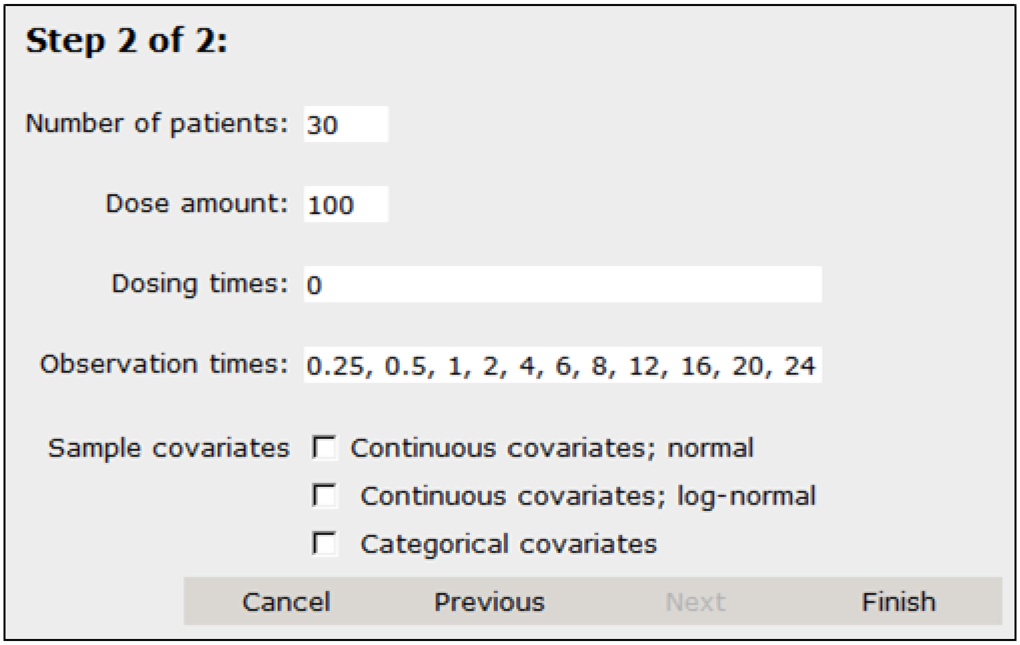
Step 1 of 2: Basic information

This wizard helps you to quickly create an R script, which creates a template dataset for quickly setting up PK(-PD) simulations from NONMEM.

R script name: sim_template.R

CSV output file: sim_001.csv

3. Enter a name for the script in the **R script name** field.
4. Enter the name for the **CSV output file** in the field.
5. Press **Next**.



Step 2 of 2:

Number of patients: 30

Dose amount: 100

Dosing times: 0

Observation times: 0.25, 0.5, 1, 2, 4, 6, 8, 12, 16, 20, 24

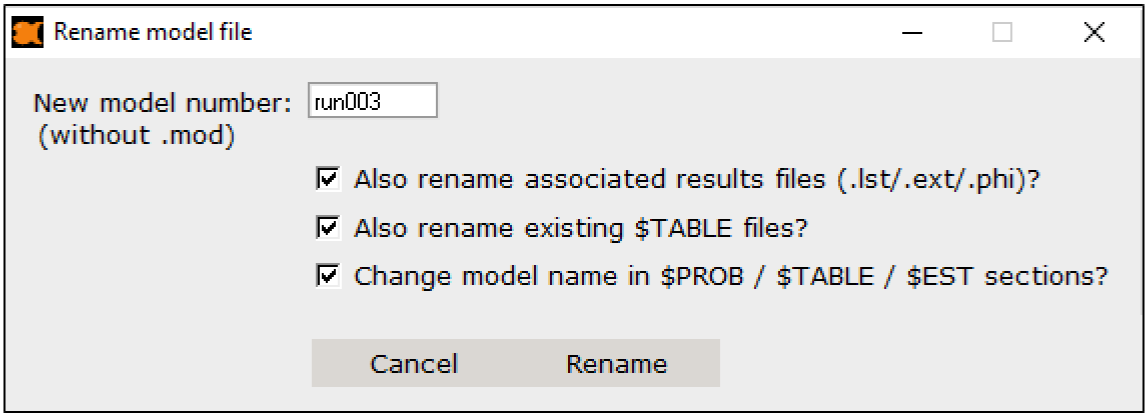
Sample covariates
□ Continuous covariates; normal
□ Continuous covariates; log-normal
□ Categorical covariates

6. Enter the specifics for the simulation, including how many patients, dose amount, when to administer the dose, and when to make the observations.
7. Check the box(es) for the types of covariates to include.
8. Press **Finish**.

The R script is saved to the current folder.

Rename a model file

1. Select the model in the list.
2. Right-click the selected model and choose **File actions > Rename** from the menu.
   Or
   Select the **Models > Rename model** menu option.
   Or
   Click ✏ in the toolbar.



New model number: (without .mod) run003

☑ Also rename associated results files (.lst/.ext/.phi)?

☑ Also rename existing $TABLE files?

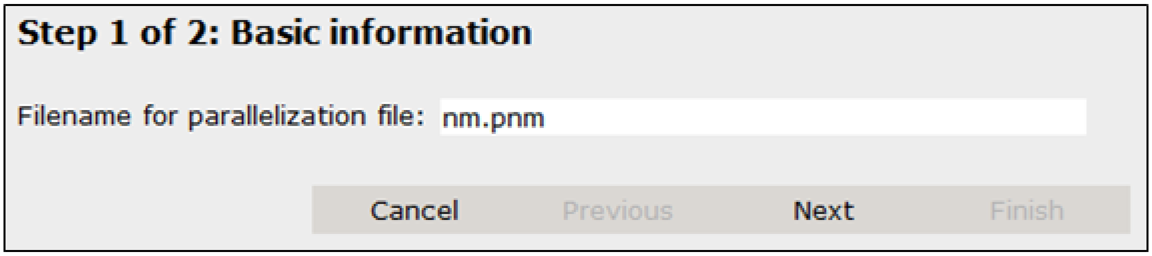
☑ Change model name in $PROB / $TABLE / $EST sections?

3. Edit the text in the **New model number** field.
4. To maintain the connection with the model using the new name, check the boxes to rename any results and table files associated with the model and to update the model name within sections of the model file itself.

Generate NM parallelization file

NONMEM 7.2 and higher allow parallelization of single runs, which requires a so-called parafile, a configuration file for the parallelization. These files can be created using the NM parallelization file wizard.

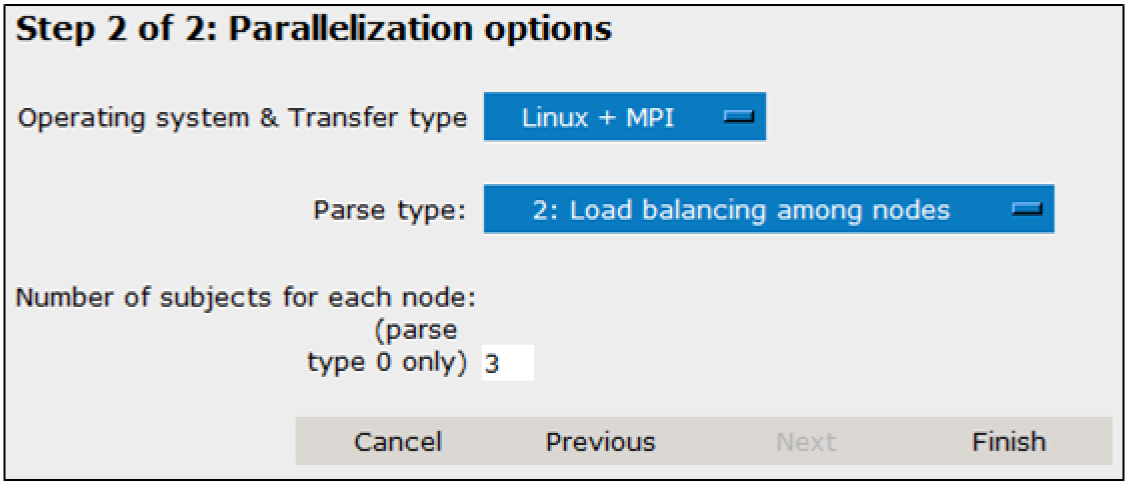
1. Select the **Tools > Wizards** menu option.
2. Choose **NM parallelization file** and click **OK**.



Step 1 of 2: Basic information

Filename for parallelization file: nm.pnm

3. Enter a name in the **Filename for parallelization file** field.
4. Press **Next**.



Step 2 of 2: Parallelization options

Operating system & Transfer type: Linux + MPI

Parse type: 2: Load balancing among nodes

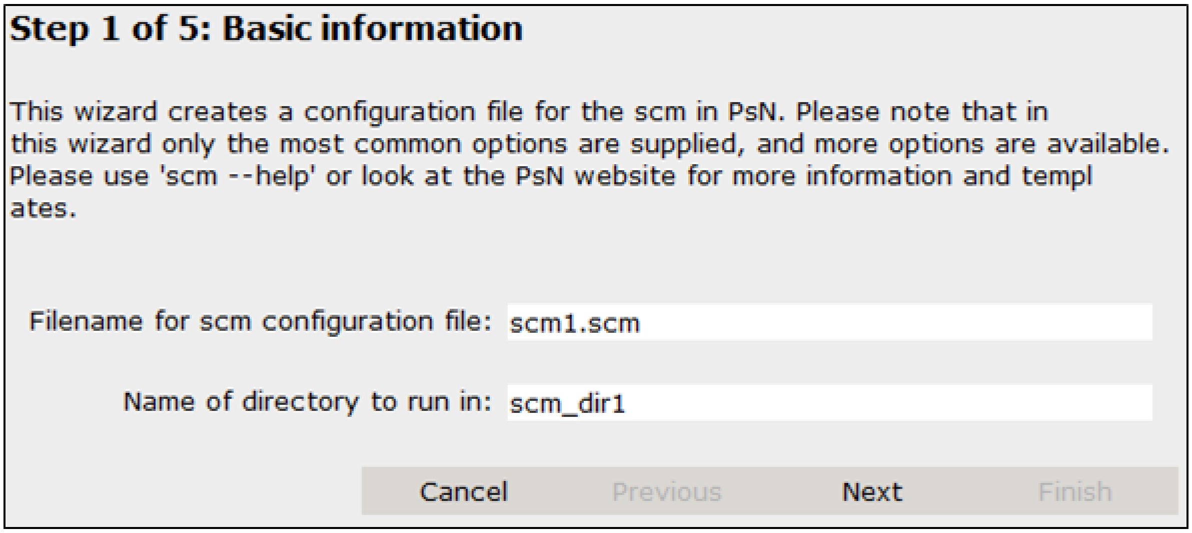
Number of subjects for each node: (parse type 0 only) 3

5. Select the desired **Operating system & Transfer type** from the pull-down.
6. Set the **Parse type** from the pull-down.
   For **Parse type** set to **0**, enter the **Number of subjects for each node** in the field.
7. Press **Finish**.

Generate SCM configuration file

The `scm` command in PsN requires a configuration file. With the SCM configuration file wizard, you can create such a file, which includes the most commonly used options. Please note that more features are available in the scm tool than are offered as option in the wizard, so it is advised to acquaint yourself with the full scm documentation.

1. Select the **Tools > Wizards** menu option.
2. Choose **SCM configuration file** and click **OK**.



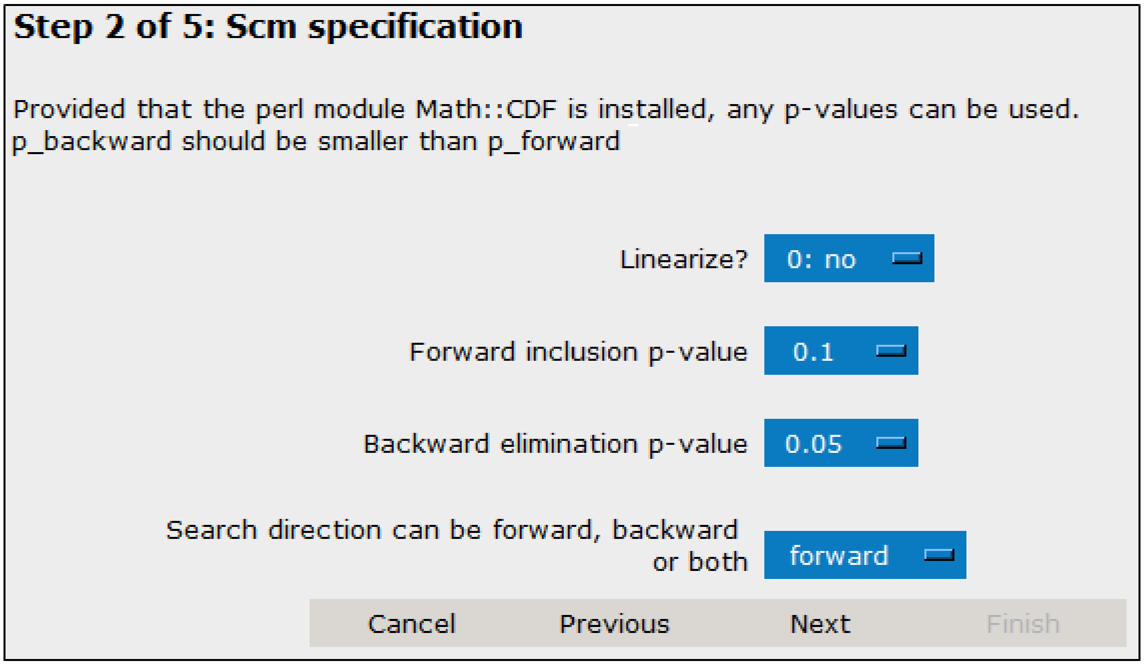
Step 1 of 5: Basic information

This wizard creates a configuration file for the scm in PsN. Please note that in this wizard only the most common options are supplied, and more options are available. Please use 'scm --help' or look at the PsN website for more information and templ ates.

Filename for scm configuration file: scm1.scm

Name of directory to run in: scm_dir1

3. Enter a name in the **Filename for scm configuration file** field.
4. Enter the desired directory in the **Name of directory to run in** field.
5. Press **Next**.



Step 2 of 5: Scm specification

Provided that the perl module Math::CDF is installed, any p-values can be used.
p_backward should be smaller than p_forward

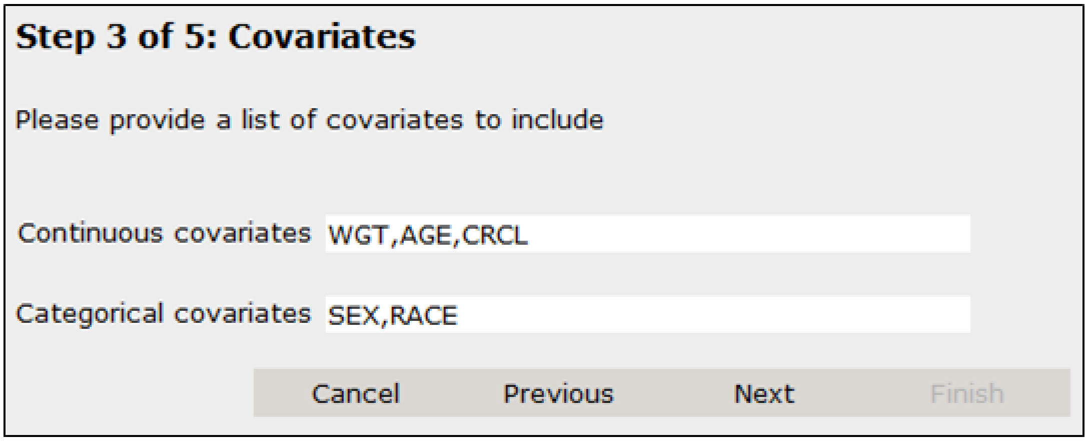
Linearize? 0: no

Forward inclusion p-value 0.1

Backward elimination p-value 0.05

Search direction can be forward, backward or both forward

6. Specify whether to **Linearize** the SCM model or not from the pull-down.
7. Select the p-value to use for inclusion during the forward inclusion step from the **Forward inclusion p-value** pull-down.
8. Select the p-value to use for exclusion during the backward elimination step from the **Backward elimination p-value** pull-down.
9. Select the direction from the **Search direction...** pull-down.
10. Press **Next**.



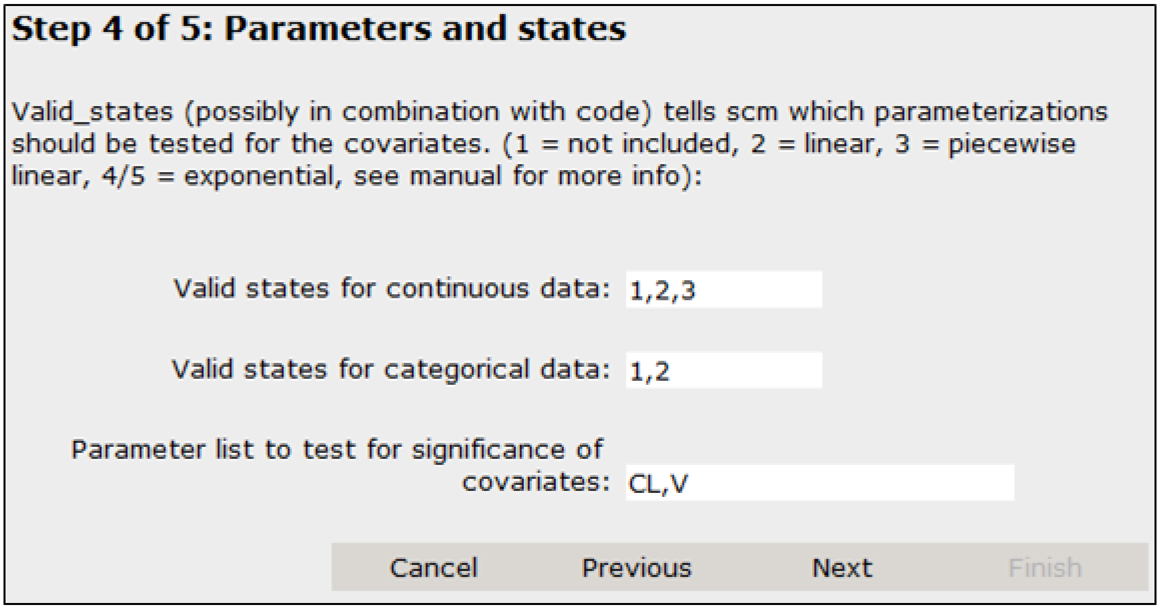
Step 3 of 5: Covariates

Please provide a list of covariates to include

Continuous covariates WGT,AGE,CRCL

Categorical covariates SEX,RACE

11. In the field, enter a comma-separated list of **Continuous covariates** to include.
12. In the field, enter a comma-separated list of **Categorical covariates** to include.
13. Press **Next**.



Step 4 of 5: Parameters and states

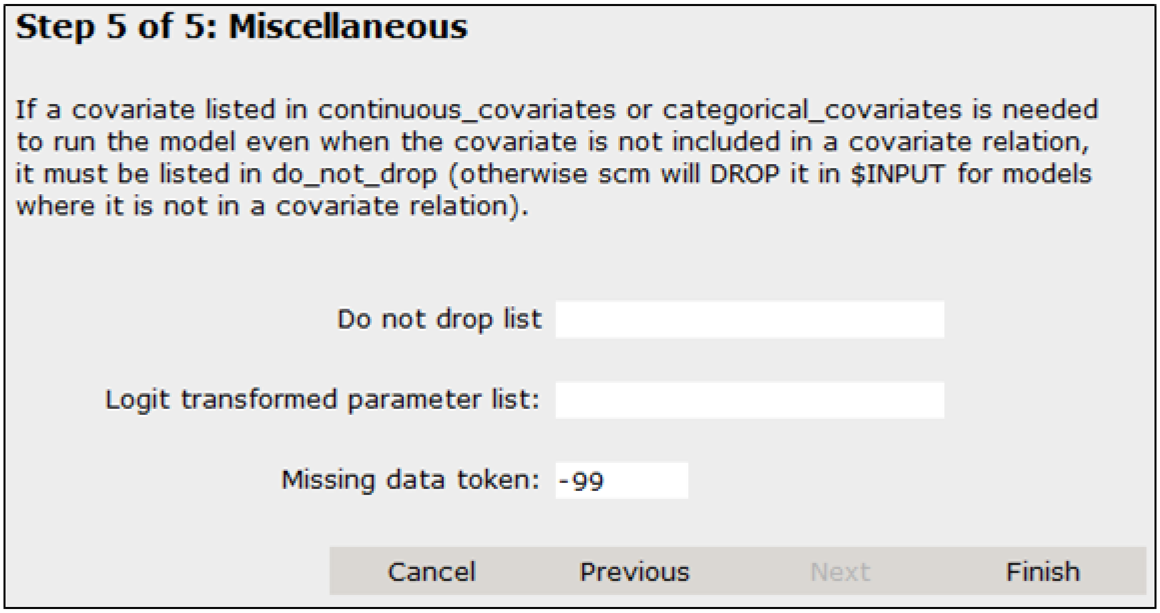
Valid_states (possibly in combination with code) tells scm which parameterizations should be tested for the covariates. (1 = not included, 2 = linear, 3 = piecewise linear, 4/5 = exponential, see manual for more info):

Valid states for continuous data: 1,2,3

Valid states for categorical data: 1,2

Parameter list to test for significance of covariates: CL,V

14. In the field, enter a comma-separated list of **Valid states for continuous data**.
15. In the field, enter a comma-separated list of **Valid states for categorical data**.
16. Enter a comma-separated list of parameters in the **Parameter list to test for significance of covariates**.
17. Press **Next**.



Step 5 of 5: Miscellaneous

If a covariate listed in continuous_covariates or categorical_covariates is needed to run the model even when the covariate is not included in a covariate relation, it must be listed in do_not_drop (otherwise scm will DROP it in $INPUT for models where it is not in a covariate relation).

Do not drop list

Logit transformed parameter list:

Missing data token: -99

18. In the **Do not drop list field**, enter a comma-separated list of covariates that should remain in the model.
19. In the **Logit transformed parameter list** field, enter a comma-separated list of parameters that involve logit transformation.
20. In the **Missing data token** field, enter the code that is used to identify missing values.
21. Press **Finish**.

The configuration file is saved in the current folder.

Execute a model

NONMEM models can be executed in any of the following ways:

- Using NONMEM through nmfe
- Using NONMEM through PsN
- Using NONMEM through Wings

Using NONMEM through nmfe

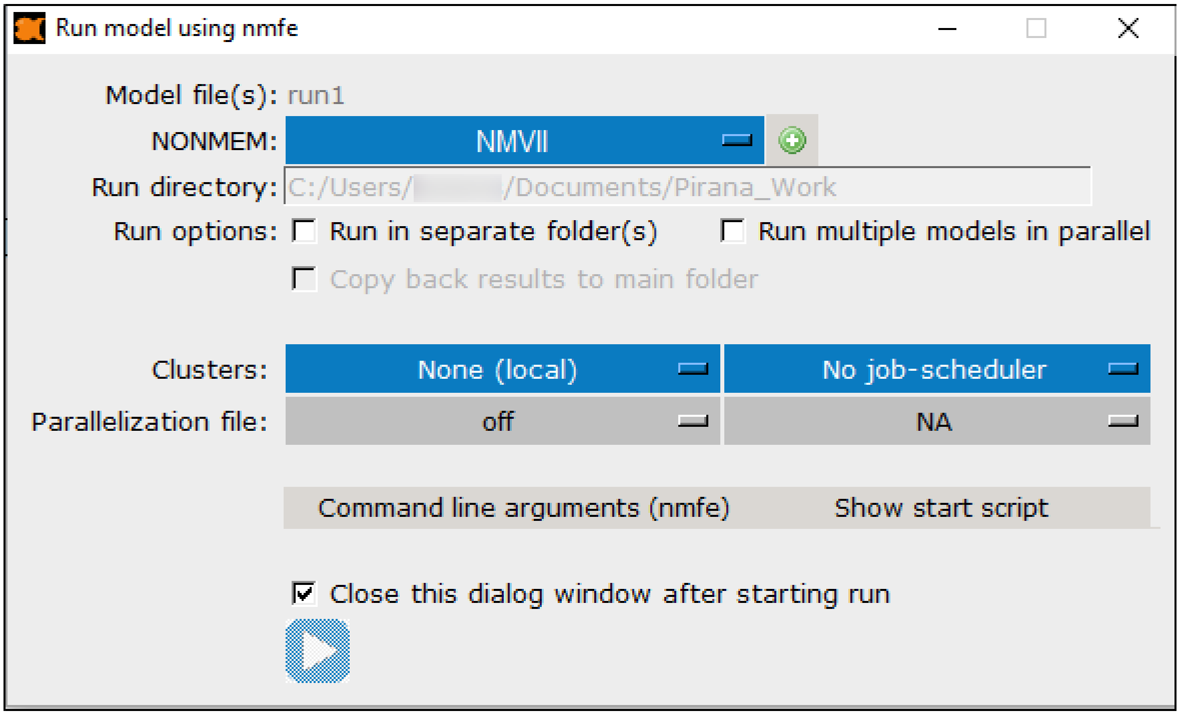
Note: To run a model through use of the nmfe script supplied with NONMEM, you have to instruct Pirana as to the location of the NONMEM installation. This is done by setting up profiles using the *Settings* dialog (see the "NONMEM" section of the *Settings* dialog description).

If you are running NONMEM through PsN, the location of NONMEM is already specified in the `psn.conf` file of PsN.

1. Select the model to run.
2. Right-click the selected model and choose **nmfe**.
   Or

   Click N in the toolbar.
   Or
   Use the **Control-R** keyboard shortcut.

The *Run* window contains a number of additional options for running the model.



Run model using nmfe

Model file(s): run1

NONMEM: NMVII

Run directory: C:/Users/[redacted]/Documents/Pirana_Work

Run options: □ Run in separate folder(s) □ Run multiple models in parallel

□ Copy back results to main folder

Clusters: None (local) No job-scheduler

Parallelization file: off NA

☑ Close this dialog window after starting run

3. Select the **NONMEM** installation from the pull-down.
   Click **+** to locate a NONMEM installation that is not listed on the pull-down.
4. The **Run directory** shows the path specified in the **Folder** field on Pirana’s main page.
5. Check the **Run in separate folder(s)** box to create individual folders, prefixed by “nmfe_”, for each model execution.
6. Check the **Run multiple models in parallel** box to allow parallelized executions.
7. Check the **Copy back results to main folder** box to bring the results from parallel runs back to the main folder.
   This option is only available when the **Run multiple models in parallel** box is checked.
8. From the **Clusters** pull-down, choose **None** to run the job locally or choose a cluster name to run the job remotely. (Refer to “Pirana and Clusters” for details on setting up a cluster in Pirana.)
9. If the chosen cluster has job schedulers associated with it, select the scheduler to use from the second pull-down.
10. Choose the configuration file to use for the parallelization.

    Parallelization configuration files can be generated using the **NM parallelization file** wizard in Pirana.

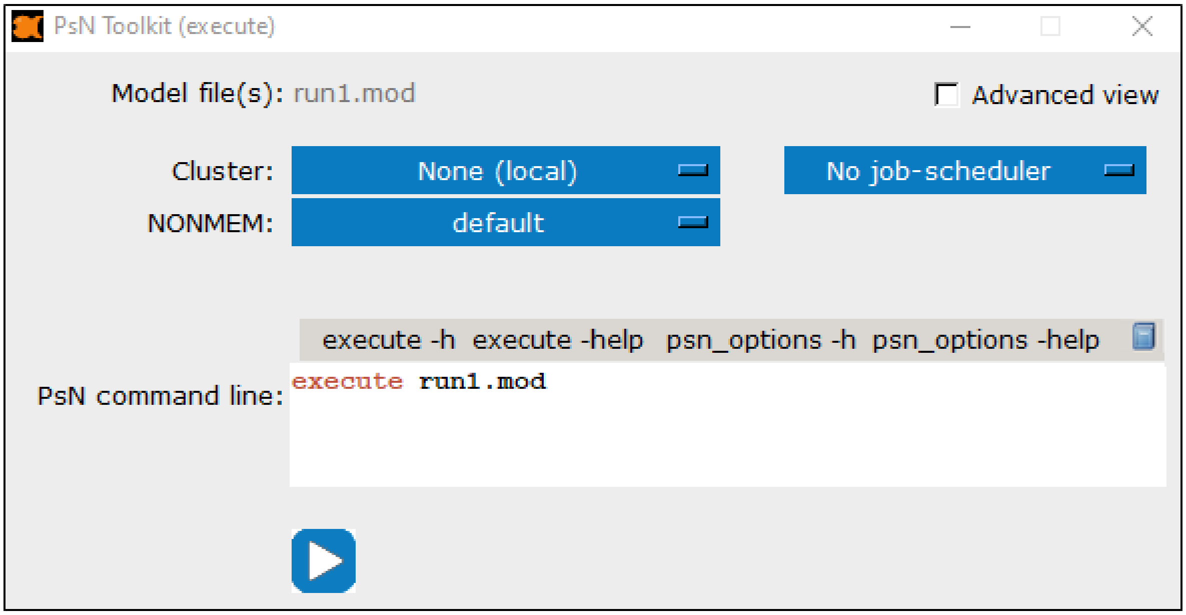
    You can also have Pirana generate the parafile on-the-fly (select **auto-MPI** or **auto-FPI** from the **Parallelization file** menu). In the *Settings* dialog, in the **NONMEM > Clusters >** ‘cluster_name’ tab, set the FPI and MPI files that Pirana generates using the respective sub-tabs. (See MPI scripts settings for parallelization and FPI scripts settings for parallelization.)

    Parallelization files can be imported from local or remote locations. Local import can be performed through the **NONMEM** option in the **Software integration** tab of the *Settings* dialog. Remote parallelization files can be imported from a cluster location defined in the **NONMEM > Remote Profiles** sub-tab of the *Settings* dialog.
11. Press **Command line arguments (nmfe)** to toggle expanding the dialog to view and optionally add nmfe command line arguments to include in the job execution.
12. Press **Show start script** to toggle expanding the dialog to view and optionally edit the model execution script and command line changes for the run options selected.
13. Uncheck the **Close this dialog window after starting run** box to keep the dialog open.
14. Click ▶ to submit the job for execution.

Using NONMEM through PsN

1. Select the model to run.
2. Right-click the selected model and choose **execute**.
   Or
   Click in the toolbar and select **execute** from the pop-up menu.
   Or
   Use the **Control-e** keyboard shortcut.

The dialog contains a number of additional options for running the model via PsN.



```
execute run1.mod
```

3. By default, Pirana will show the dialog in ***simple view***, check the **Advanced View** to see additional options:
   – Specify R scripts to execute before/after the model run.
4. From the **Clusters** pull-down, choose **None** to run the job locally or choose a cluster name to run the job remotely. (Refer to "Pirana and Clusters" for details on setting up a cluster in Pirana.)
5. If the chosen cluster has job schedulers associated with it, select the scheduler to use from the second pull-down.
6. Select the **NONMEM** installation from the pull-down.
7. In the **PsN command line** field, type the PsN commands to use to execute the job.

   Select one of the buttons above the field to view argument information:
   vpc -h is shorthand help
   vpc -help is expanded help
   psn_options -h is the command to list all common PsN options
   psn_options -help is the expanded list of all common PsN options

   Click to view a history of the command's usage in a separate dialog. You can select one from the list and press **Use command** to copy the command and arguments back to the *PsN Toolkit* dialog. In the PsN configuration file (`psn.conf`), a list of default arguments can be supplied as well, so commonly used arguments do not have to be repeated on the command line.

8. Click ▶ to submit the job for execution.

PsN creates a subfolder in which the run is executed. After the NONMEM run finishes, PsN will copy back the results files to the main folder.

Note that Pirana does not automatically detect that new results are available, so you should press the refresh button to load the results into the *Pirana* window. To show the folders that PsN has created in the main window, select either **PsN folders** or **All folders** from the folder selection menu.

Using the **PsN** options in the Software integration tab in the *Settings* dialog, you can define the default command line parameters for most PsN functions.

Using NONMEM through Wings

Note: Support for Wings must be turned on in the Pirana settings (**Settings > General > Enable Pirana support for WFN**, then restart Pirana).

1. Select the model to run.
2. Right-click the selected model and choose **Run via Wings**.
   Or
   Click the corresponding toolbar icon.

On Windows, Pirana is capable of invoking the WFN-commands `nmgo` and `nmbs`, for run execution and bootstrapping, respectively. Since WFN does not support multiple model files to be processed by its commands, when multiple models are selected, only the first model file is executed. When the WFN method is selected, two parameter specification bars will become visible. In the upper entry, run parameters can be specified (e.g., for the bootstrap: `1 100` to specify a bootstrap with 100 replicates). The lower parameter bar specifies command-line parameters used when starting `WFN.bat` (e.g., `g77 std` for specifying the compiler and the NONMEM version to be used). Use the **NONMEM** options in the Software integration tab in the *Settings* dialog to specify these command-line parameters.

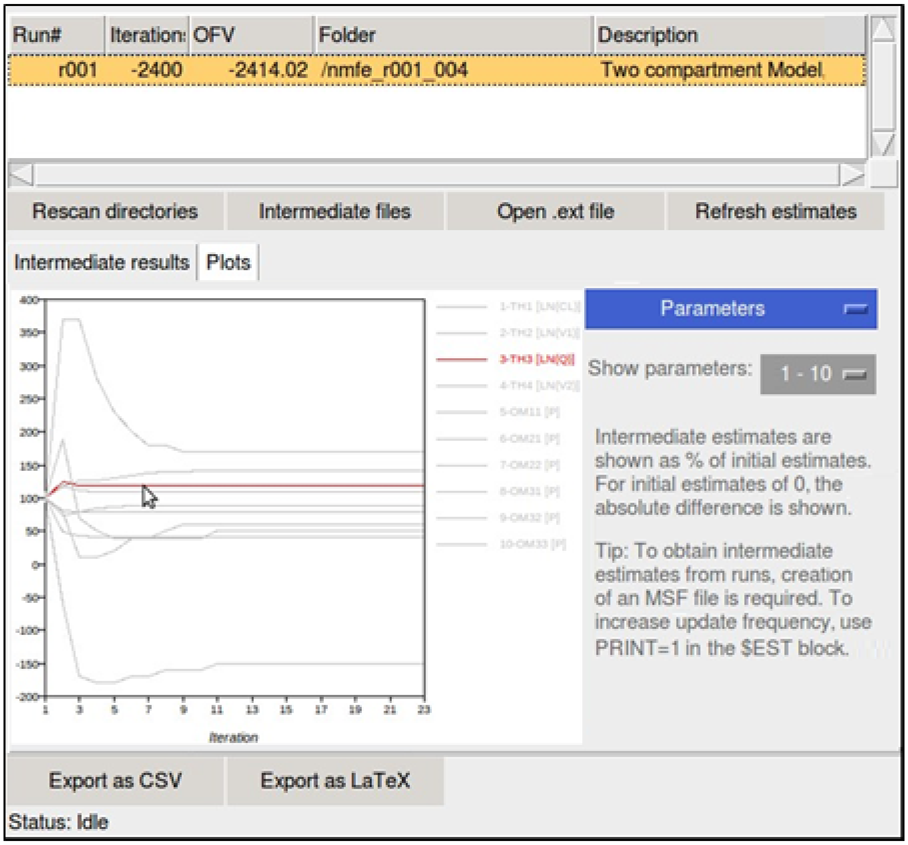
Note: When executing runs through WFN, Pirana creates a temporary batch-file in the current directory that starts `WFN.bat` to load the necessary environment variables, after which `nmgo` is started with the model-file and parameters specified.

View intermediate results of active runs

When editing and running models, Pirana can automatically backup all versions of controlstreams and result files. When this **Auto-backup models & results** option is activated in the *Setting* dialog (see General settings), intermediate versions of models and results are saved in the `pirana_backup` subdirectory of the active project, every time a model is changed, or when a new results files is found.

During a run, intermediate results can be accessed via the ⌛ icon in the toolbar, or by selecting the **View > Intermediate results of active runs** menu option. In the *Progress* dialog, there is an **Intermediate results** tab. Details of the run and (if applicable) parameter estimates can be reviewed.

By clicking on a run in the list, Pirana will parse the intermediate files and show the intermediate parameter estimates in the table and the plot. In the plot, you can choose to show either the gradients (if a gradient method is used), the intermediate estimates, or the objective function value (OFV). Make sure that you specify `PRINT=1` and `MSFO=xxxxx` in the $ESTIMATION record, to be able to obtain regularly updated intermediate estimates. From the submenu, signals can be sent to currently running model processes, such as `stop` and `next iteration`.



Run#	Iteration	OFV	Folder	Description
r001	-2400	-2414.02	/nmfe_r001_004	Two compartment Model

Show parameters: 1 - 10

Intermediate estimates are shown as % of initial estimates. For initial estimates of 0, the absolute difference is shown.

Tip: To obtain intermediate estimates from runs, creation of an MSF file is required. To increase update frequency, use PRINT=1 in the $EST block.

Status: Idle

Click **Export as CSV** or **Export as LaTeX** to send the intermediate results to a file.

Model translation

1. Select a model in the list.
2. Right-click the selected model and choose **Translate model** from the menu.
   Or

   Click 🖧 in the toolbar.
3. Select the format from the pop-up list:
   To NONMEM
   To R
   To Matlab
   To Berkely Madonna
4. Enter a name for the new file in the field (click 📁 to browse to a different location).
5. Press **Translate**.

To NONMEM

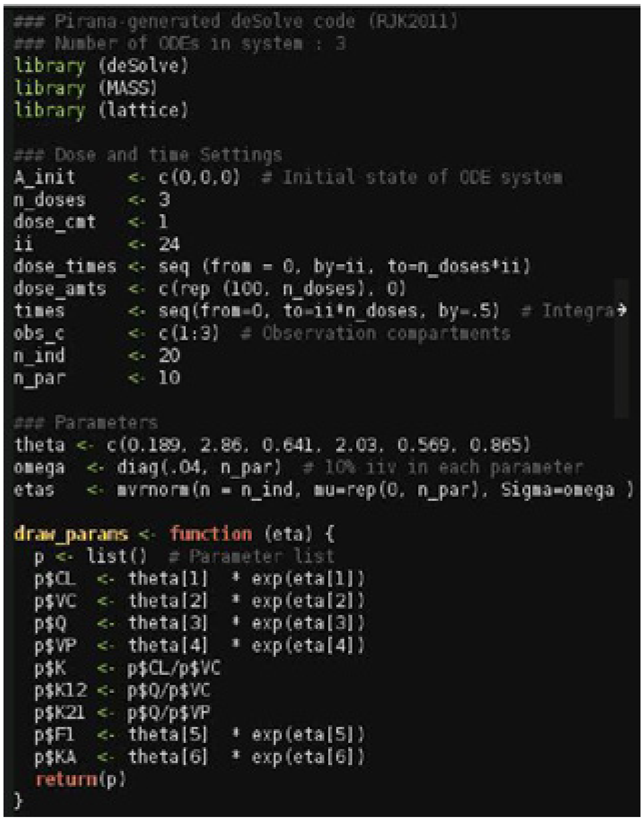
Pirana includes translators that will convert specific parts of NONMEM code to alternate NONMEM code.

- **ADVAN 1-6 to $DES**: This option translates NM-TRAN models written in ADVAN routine (1-6) to ordinary differential equations (ODEs).
- **MU-referencing thetas-etas**: Converts models written using normal- or log-normal ηs, e.g.
  ```
     CL = THETA(1) * EXP(ETA(1))
  ```
  will be converted into:
  ```
     MU_1 = LOG(THETA(1)) ; ** MU-referenced by Pirana
     CL = EXP(MU_1+ETA(1))
     ; Original equation: CL = THETA(1) * EXP(ETA(1))
  ```
- **$DES to difference equations**: For some models written in ODEs, writing some parts of the model in difference equations can considerably reduce computational burden, while maintaining parameter precision (Petersson, K.J., et al. *J Pharmacokinet Pharmacodyn*. 2010 Oct. 37(5):493-506). Using this option translates all code written in $DES, other than the *dA/dt* system, to $PK and adds some required code (using *MTIME*).

To R

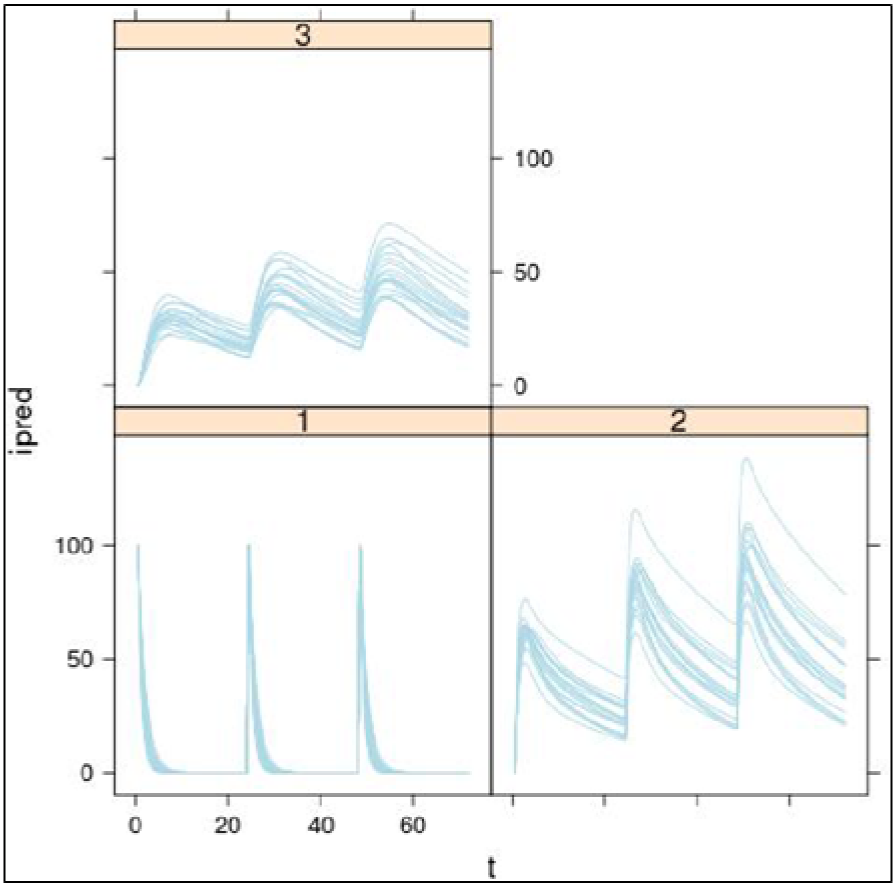
The R translation options extract the parameter estimates for the structural model (θ) and also the between subject variability matrix (Ω). A multi-dose simulation is automatically implemented (Pirana currently does not read in the dataset to extract dosing information). No residual error model is currently implemented in the R translators, but can be added by the user.

- **PKPDSim**: This package is not available on CRAN, but may be sourced from GitHub, e.g., within R, use:
  ```
  library(devtools)
  install_github("ronkeizer/PKPDsim")
  library(PKPDsim)
  ```
- **deSolve**: The R code generated using this option is automatically loaded in the defined R interface.
  An example of generated R code is depicted in the following image.



```
### Pirana-generated deSolve code (RJK2011)
### Number of ODEs in system : 3
library (deSolve)
library (MASS)
library (lattice)

### Dose and time Settings
A_init     <- c(0,0,0)  # Initial state of ODE system
n_doses    <- 3
dose_cmt   <- 1
ii         <- 24
dose_times <- seq (from = 0, by=ii, to=n_doses*ii)
dose_amts  <- c(rep (100, n_doses), 0)
times      <- seq(from=0, to=ii*n_doses, by=.5)  # Integra
obs_c      <- c(1:3)  # Observation compartments
n_ind      <- 20
n_par      <- 10

### Parameters
theta <- c(0.189, 2.86, 0.641, 2.03, 0.569, 0.865)
omega  <- diag(.04, n_par)  # 10% iiv in each parameter
etas   <- mvrnorm(n = n_ind, mu=rep(0, n_par), Sigma=omega )

draw_params <- function (eta) {
  p <- list()  # Parameter list
  p$CL  <- theta[1]  * exp(eta[1])
  p$VC  <- theta[2]  * exp(eta[2])
  p$Q   <- theta[3]  * exp(eta[3])
  p$VP  <- theta[4]  * exp(eta[4])
  p$K   <- p$CL/p$VC
  p$K12 <- p$Q/p$VC
  p$K21 <- p$Q/p$VP
  p$F1  <- theta[5]  * exp(eta[5])
  p$KA  <- theta[6]  * exp(eta[6])
  return(p)
}
```

An example of associated simulation output for the **deSolve**-generated R code is depicted below.



To Matlab

The **Matlab** translation options extract the parameter estimates for the structural model (θ) only (the between subject variability matrix is not extracted). No residual error model is currently implemented in the R translators, but can be added by the user.

- **to ODEs**
- **to PopED files**: Porting the model structure to PopED allows evaluation of optimal study designs (OD). This option creates the necessary files for PopED execution, however the details of the design and other optimization settings still need to be provided.

The generated Matlab code is automatically loaded in the defined code editor.

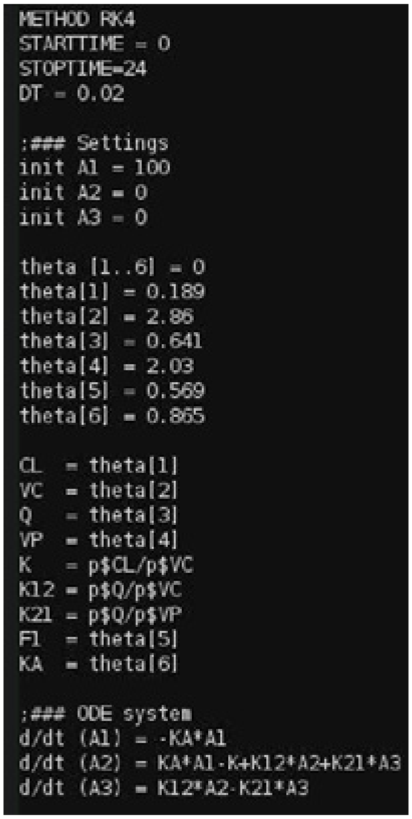
To Berkely Madonna

The **Berkely Madonna** translation option extracts the parameter estimates for the structural model (θ) and also the between subject variability matrix (Ω). No residual error model is currently implemented in the R translators, but can be added by the user.

Check the **Include between-subject variability** box to consider variability between subjects. If checked, then choose the type of sampling to perform during the between-subject variability computations: **Univariate sampling** or **Multivariate sampling**.

The generated Berkeley Madonna code is automatically loaded in the defined code editor.

An example of generated Berkely Madonna code is depicted in the following image.

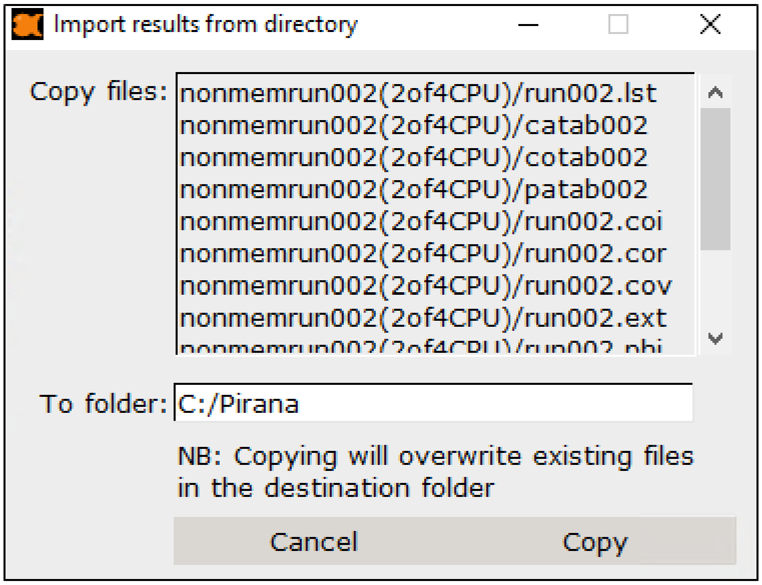


```
METHOD RK4
STARTTIME = 0
STOPTIME=24
DT = 0.02

;### Settings
init A1 = 100
init A2 = 0
init A3 = 0

theta [1..6] = 0
theta[1] = 0.189
theta[2] = 2.86
theta[3] = 0.641
theta[4] = 2.03
theta[5] = 0.569
theta[6] = 0.865

CL  = theta[1]
VC  = theta[2]
Q   = theta[3]
VP  = theta[4]
K   = p$CL/p$VC
K12 = p$Q/p$VC
K21 = p$Q/p$VP
F1  = theta[5]
KA  = theta[6]

;### ODE system
d/dt (A1) = -KA*A1
d/dt (A2) = KA*A1-K*K12*A2+K21*A3
d/dt (A3) = K12*A2-K21*A3
```

Import results from a model directory

1. Select a model in the list.
2. Right-click the selected model and choose **File actions > Import results from directory** from the menu.
   Or

   Click in the toolbar.



Import results from directory

Copy files:

```
nonmemrun002(2of4CPU)/run002.lst
nonmemrun002(2of4CPU)/catab002
nonmemrun002(2of4CPU)/cotab002
nonmemrun002(2of4CPU)/patab002
nonmemrun002(2of4CPU)/run002.coi
nonmemrun002(2of4CPU)/run002.cor
nonmemrun002(2of4CPU)/run002.cov
nonmemrun002(2of4CPU)/run002.ext
nonmemrun002(2of4CPU)/run002.phi
```

To folder: C:/Pirana

NB: Copying will overwrite existing files in the destination folder

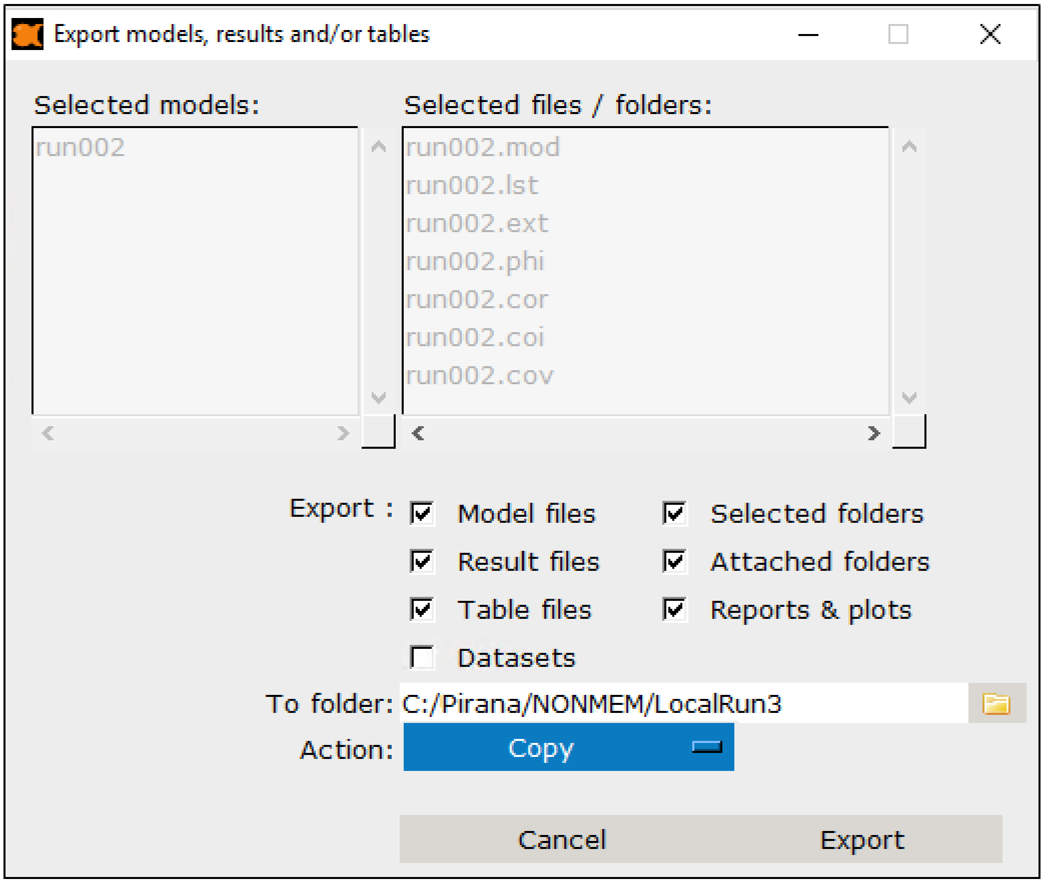
3. Select the files in the **Copy files** list that are to be copied from the model's results directory.
4. In the **To folder** field, enter the full path to the destination folder (click 📁 to navigate to the location).
5. Click **Copy**.

Note: Existing files in the destination folder will be overwritten by copied files with the same name.

Export model items

1. Select the model(s) for exporting.
2. Right-click the selected model and choose **File actions > Export to (sub)folder** from the menu. Or

   Click 📁 in the toolbar.



Export models, results and/or tables

Selected models:

run002

Selected files / folders:

run002.mod
run002.lst
run002.ext
run002.phi
run002.cor
run002.coi
run002.cov

Export : ☑ Model files ☑ Selected folders
☑ Result files ☑ Attached folders
☑ Table files ☑ Reports & plots
☐ Datasets

To folder: C:/Pirana/NONMEM/LocalRun3

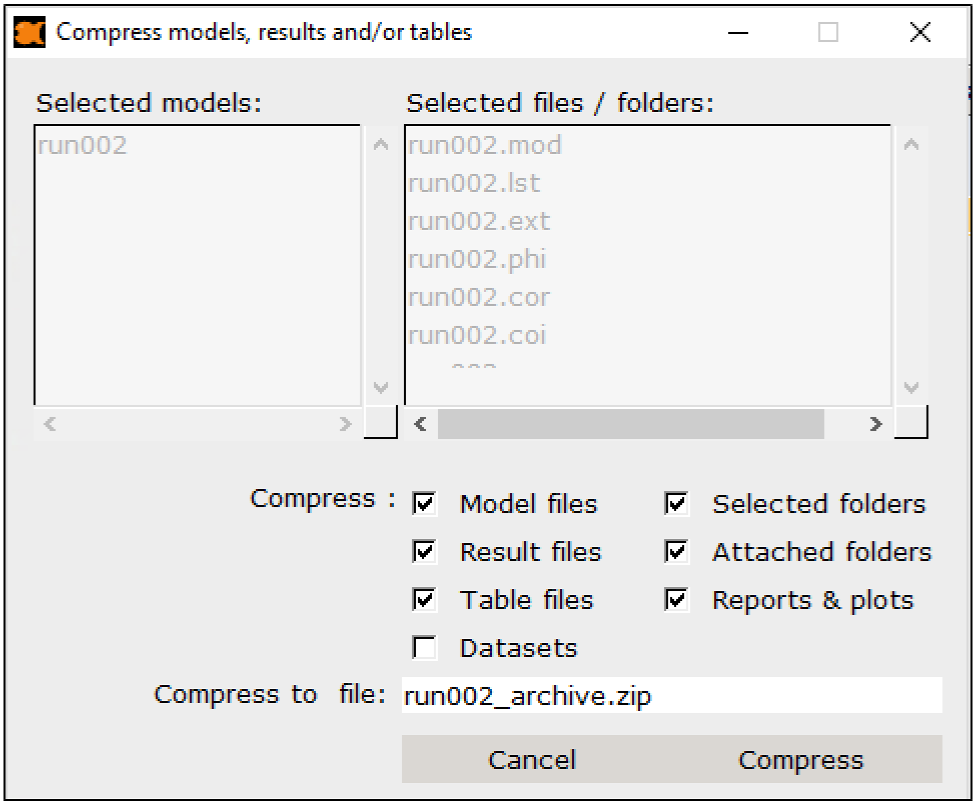
Action: Copy

3. Check the boxes for the types of items to export.
   The **Selected files/folders** list will highlight the items as you check the boxes.
4. In the **To folder** field, enter the full path to the destination folder (click 📁 to navigate to the location).
5. In the **Action** pull-down, select **Copy** to create a copy of the items in the new location or select **Move** to the items to the new location.
6. Click **Export**.

Create zip file of model items

As a final step in model development, you may want to bundle the final model file, the associated result files, any attached folders, output files, and VPC folder into a zip file.

1. Select the model in the list.
2. Right-click the selected model and choose **File actions > Compress to zipfile** from the menu.



Selected models:
run002

Selected files / folders:
run002.mod
run002.lst
run002.ext
run002.phi
run002.cor
run002.coi

Compress : ☑ Model files ☑ Selected folders
☑ Result files ☑ Attached folders
☑ Table files ☑ Reports & plots
☐ Datasets

Compress to file: run002_archive.zip

3. Check the boxes for the types of items to include in the compressed file.
   The **Selected files/folders** list will highlight the items as you check the boxes.
4. In the **To folder** field, enter the full path to the destination folder (click 📁 to navigate to the location).
5. Click **Compress**.

Diagnostic graphs with Xpose

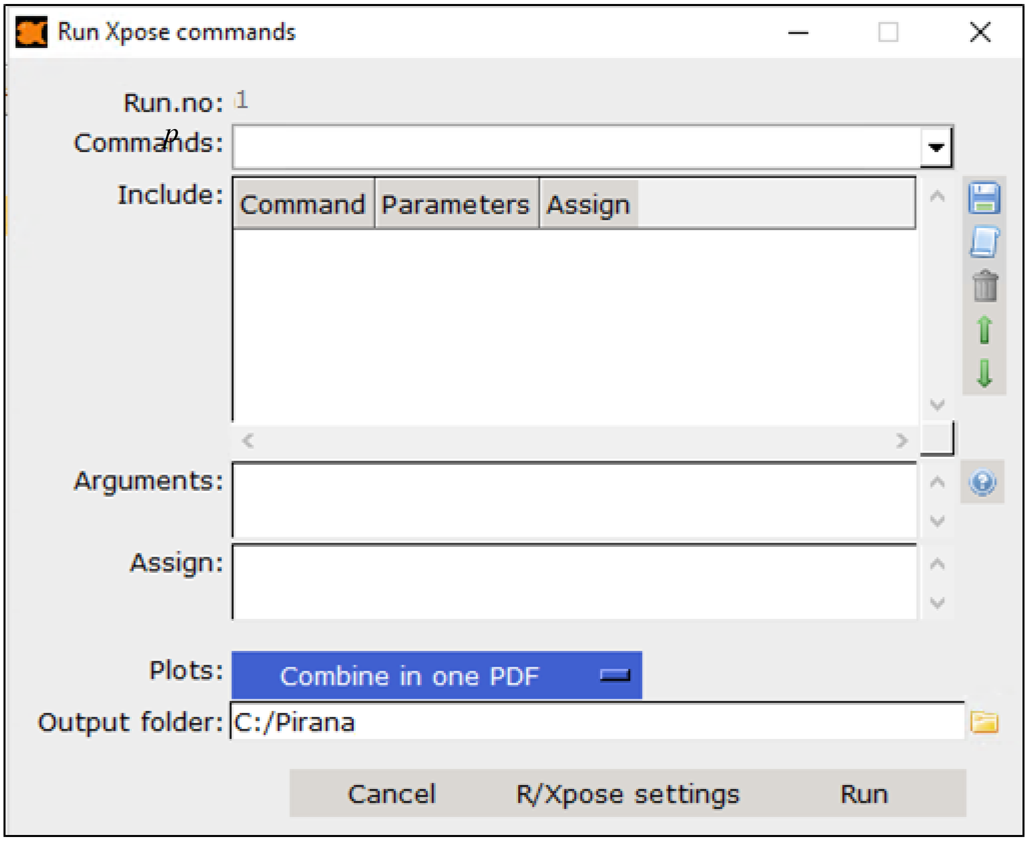
Before Xpose diagnostic graphics can be generated, the model first needs to be executed while generating output tables in a specific format and naming.

Briefly, for a model named `run10.mod`, output tables such as `sdtab10` (observations/predictions), `patab10` (parameters), `cotab10` (continuous covariates), and `catab10` (categorical covariates) should be generated, with the NOPRINT and ONEHEADER options.

For more information on how to generate Xpose-ready $TABLE output files, refer to the Xpose manual.

1. Click X in the toolbar.
   Or
   From the right-click menu of a selected model that has been executed, choose **Xpose > Xpose**

GUI.
Or
Select **Results > Xpose > Xpose GUI** from the main menu.



Run Xpose commands

Run.no: 1

Commands:

Include:

Command	Parameters	Assign

Arguments:

Assign:

Plots: Combine in one PDF

Output folder: C:/Pirana

2. In the dialog, use the **Commands** menu to add Xpose plots to the **Included** list.

   – Save and load the commands by clicking 💾 and 📄, respectively. This can be useful e.g. for standardized report generation.

   – Select a command in the list and click 🗑 to remove it.

   – Use the up and down arrows to reorder the commands.

3. Enter additional **Arguments** for each Xpose command in the field.

   A reference to possible arguments is provided under the ❓ sign.

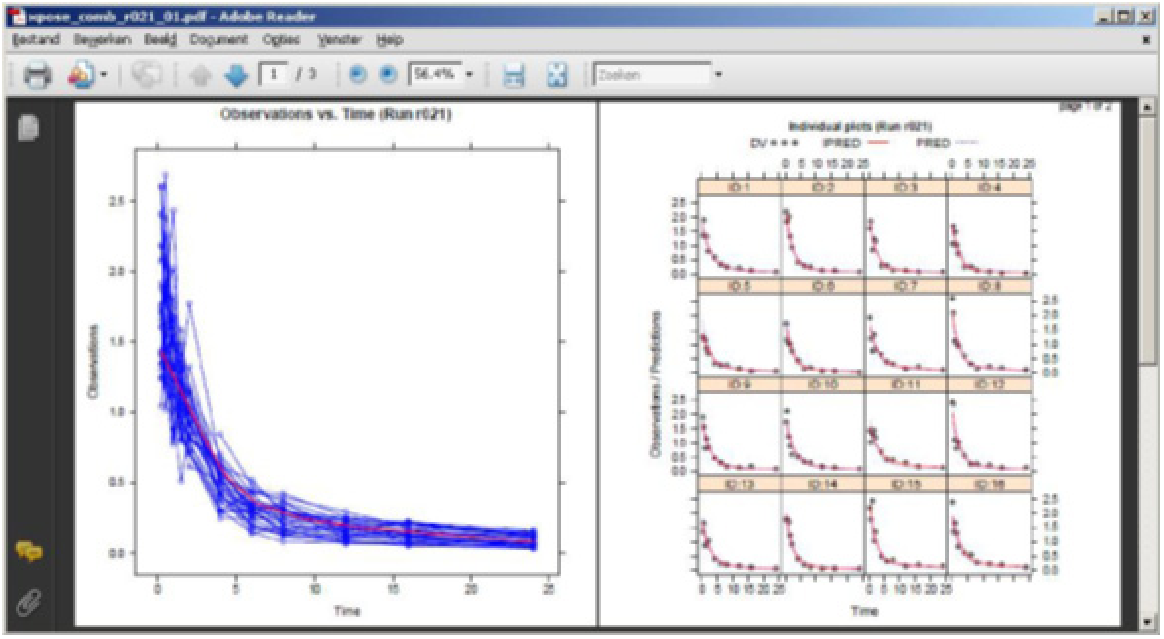
   The general plotting arguments for PDF and PNG (e.g., width=10, height=8) can be specified in the *Settings* dialog (see R/Xpose setup).

4. Select an output format from the **Plots** pull-down.
   – **Combine in one PDF**
   – **Generate PNG**
   – Generate the **R-code only**
   – Generate **Sweave code** for LaTeX documents

5. Enter the location for the generated plots in the **Output folder** field or click the 📁 to browse to the location.

6. Click **R/Xpose Settings** to display R/Xpose setup (sub-tab of Software Integration) in the *Settings* dialog.
7. Click **Run** to generate the plots.

If output was directed to a PDF file, the PDF will be opened automatically once it is generated.


xpose_comb_r021_01.pdf - Adobe Reader
Bestand Bewerken Beeld Document Opties Venster Help
1 / 3
56.4%
Zoeken
Observations vs. Time (Run r021)
Observations
Time
Individual plots (Run r021)
DV ● ● ●
IPRED
PRED
ID:1
ID:2
ID:3
ID:4
ID:5
ID:6
ID:7
ID:8
ID:9
ID:10
ID:11
ID:12
ID:13
ID:14
ID:15
ID:16
Observations / Predictions
Time

Generate Xpose graphs through R

Alternatively, it is possible to automatically open the text-based Xpose menu in R from within Pirana.

1. Select the executed model in the model overview list.
2. Right-click the selected model and choose **Xpose > Start Xpose menu in R**.
   Or
   Select **Results > Xpose > Start Xpose menu** from the main menu.

The Xpose menu is started in R and the associated table files are loaded into Xpose, where graphs may be generated. Refer to the Xpose4 help for additional information.

NONMEM VPC plots with PsN and Xpose

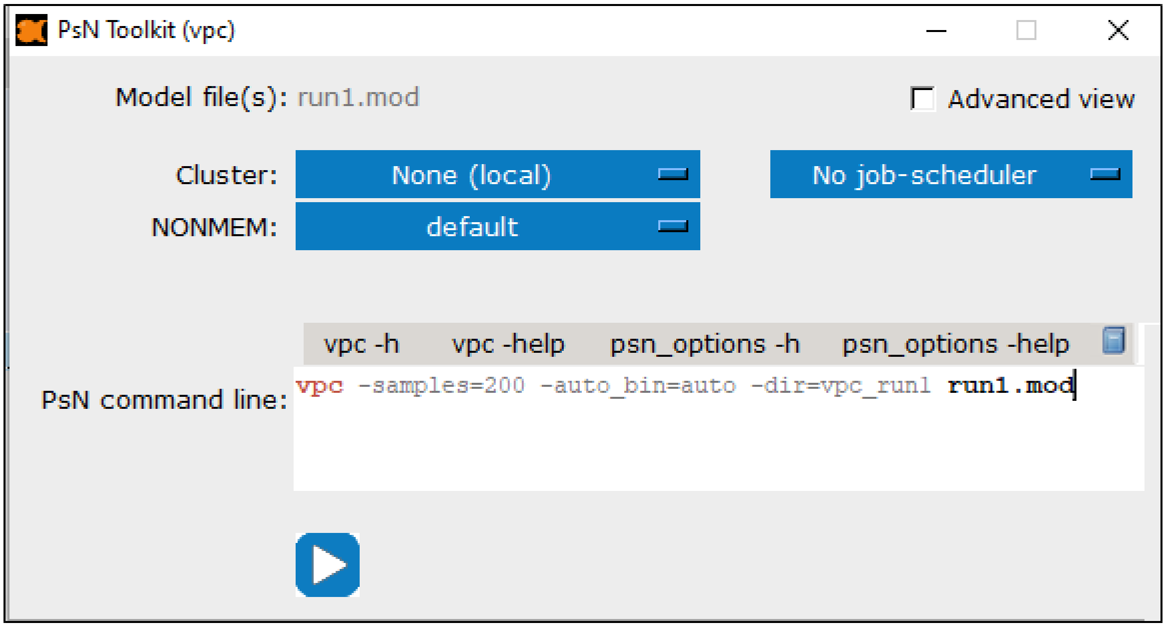
In Pirana, generate VPC data using PsN and then use that data to create VPC plots with Xpose using different plotting options.

- Generating data for the VPC
- Plotting the VPC data using Xpose
- Plotting the VPC data using direct scripts

Generating data for the VPC

1. Select the model for which a VPC should be created.
2. Right-click the selected model and choose **PsN other > Model diagnostics > vpc**.
   Or

   Click in the toolbar and select **Model diagnostics > vpc** from the pop-up menu.



PsN Toolkit (vpc)

Model file(s): run1.mod

□ Advanced view

Cluster: None (local) No job-scheduler

NONMEM: default

PsN command line:

```
vpc -samples=200 -auto_bin=auto -dir=vpc_run1 run1.mod
```

3. In the dialog, select the **Cluster** name on which to execute the VPC run from the pull-down.
4. Select the cluster type from the second pull-down.
5. Choose the **NONMEM** version from the pull-down.
6. Enter the command for creating the vpc in the **PsN command line** field.

   The vpc command takes many arguments which alters the way the vpc is calculated, e.g. you can specify stratifications, binning, dependent variable etc. When entering arguments, make sure to separate them by a space and start each argument with a '-'.

   Select one of the buttons above the field to view argument information:
   vpc -h is shorthand help
   vpc -help is expanded help
   psn_options -h is the command to list all common PsN options
   psn_options -help is the expanded list of all common PsN options

Click to view a history of the command's usage in a separate dialog. You can select one from the list and press **Use command** to copy the command and arguments back to the *PsN Toolkit* dialog. In the PsN configuration file (`psn.conf`), a list of default arguments can be supplied as well, so commonly used arguments do not have to be repeated on the command line.

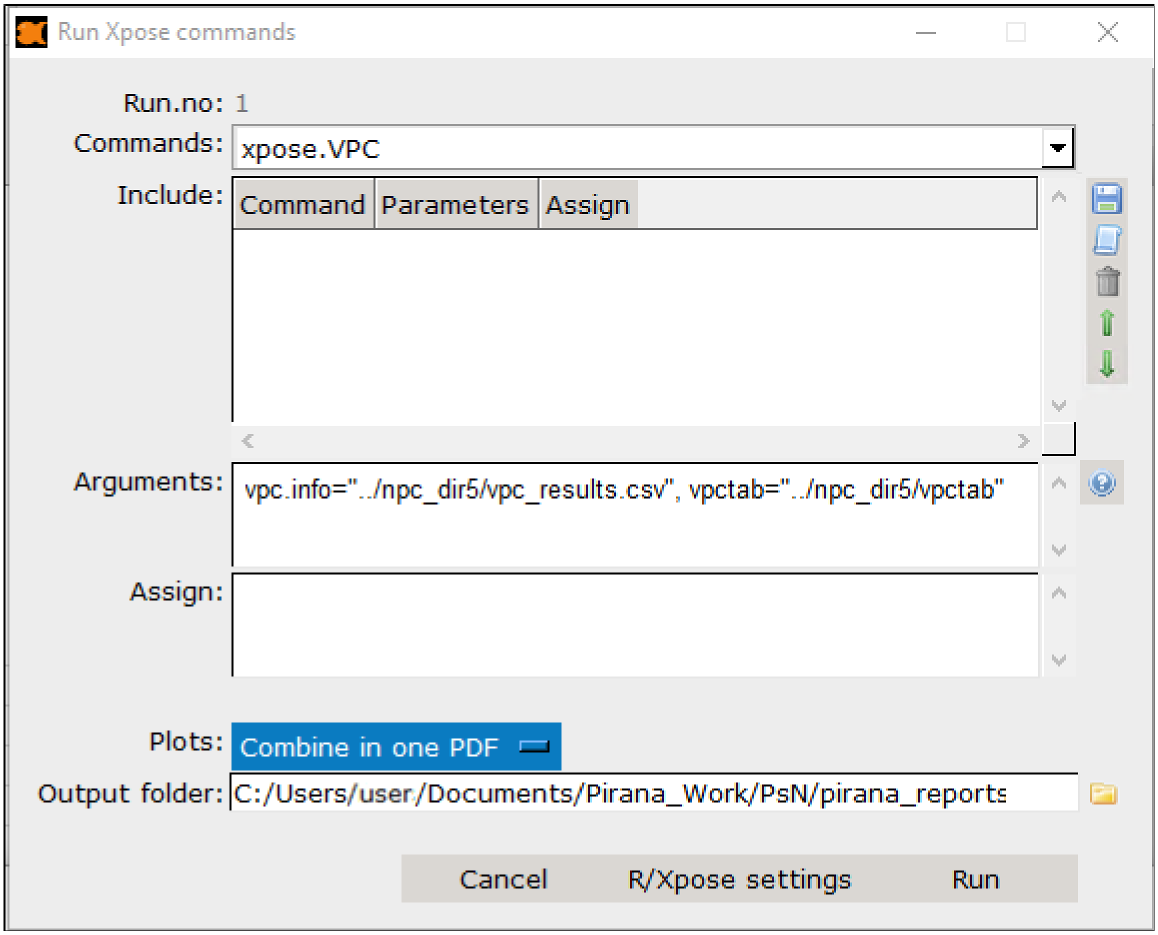
7. When all arguments for the VPC dataset have been defined correctly, click ▶ to execute.

After successful completion of the VPC calculation, a new folder is added in the *Pirana* window list. If you do not see the folder, click to refresh the list (or right-click a selected model in choose **Model > Refresh model info** from the menu). Also, make sure that the folder filter is set to **PsN folders** or **All folders**.

Plotting the VPC data using Xpose

This section provides steps to generate a VPC plot, however Xpose offers many graphing options. See "Diagnostic graphs with Xpose" for a more information about the *Run Xpose commands* dialog or refer to "Generate Xpose graphs through R" for an alternative mechanism.

1. Select the model for which the VPC was executed.
2. Right-click the selected model and choose **Xpose > Xpose GUI** in the menu.
   Or
   Select **Results > Xpose > Xpose GUI**.
   Or
   Click in the toolbar.
3. In the *Run Xpose commands* dialog, select **xpose.VPC** from the Commands pull-down.



Run.no: 1

Commands: xpose.VPC

Include:

Arguments: vpc.info="../npc_dir5/vpc_results.csv", vpctab="../npc_dir5/vpctab"

Assign:

Plots: Combine in one PDF

Output folder: C:/Users/user/Documents/Pirana_Work/PsN/pirana_reports

4. Enter arguments for Xpose.VPC in the Arguments field.

   The Xpose.VPC help files may be accessed using ❓.

5. Select the plots to generate from the **Plots** pull-down.
   Combine in one PDF
   Create in R environment
   Save as PNGs
   Save as separate PDFs

6. Specify the location for the output in the **Output folder** field (click 📁 to navigate to and select the folder).

7. Press **Run** to execute.

Tweaking the VPC plot

These VPCs may be further optimized by adjusting the many Xpose arguments. Please check out the Xpose VPC help files (which may be accessed from the *Run Xpose command* dialog, or go to the Xpose website for more information).

Plotting the VPC data using direct scripts

There are direct scripts available to plot VPC results in a few themes.

1. Select the vpc output folder, e.g., vpc_run1.
2. Select the **Scripts** tab on the right.
3. Under **NONMEM**, expand **PsN**.

   There are two options: **Plot_vpc_results.R** and **Plot_vpc_results_bins.R**.

4. Expand **VPC**.

   There are four options: **vpc_ggplot_bins_style1.R**, **vpc_ggplot_bins_style2.R**, **vpc_ggplot_smooth_style1.R**, and **vpc_ggplot_smooth_style2.R**.

5. Expand **Xpose4 > VPC**.

   There are four options: **basic.R**, **basic_logY.R**, **style_1.R**, and **style_1_nodata.R**.

6. Select the script(s) and click in the Scripts tab toolbar to run them directly.
   Or
   Right-click a selected script and choose **Open script in Rstudio** from the menu to edit.

Automated Modeling Workflow Example

An automated modeling workflow for NONMEM models alleviates the burden on modeling scientist by removing the repetitive task of running and evaluating many candidate models, standardizes the model development between modelers, and standardize the results reported from such an analysis ultimately leading to higher quality of PopPK analyses (Schmidt et al. JPKPD 2014 Aug). In Pirana (version >= 2.10), such a workflow is made available, and in this tutorial we will walk through an example of an automated population PK analysis.

For this tutorial, we will use the template model library that is provided with Pirana, and a (simulated) dataset of an iv-administered drug also provided with Pirana (demo.csv).

- New analysis
- Reports

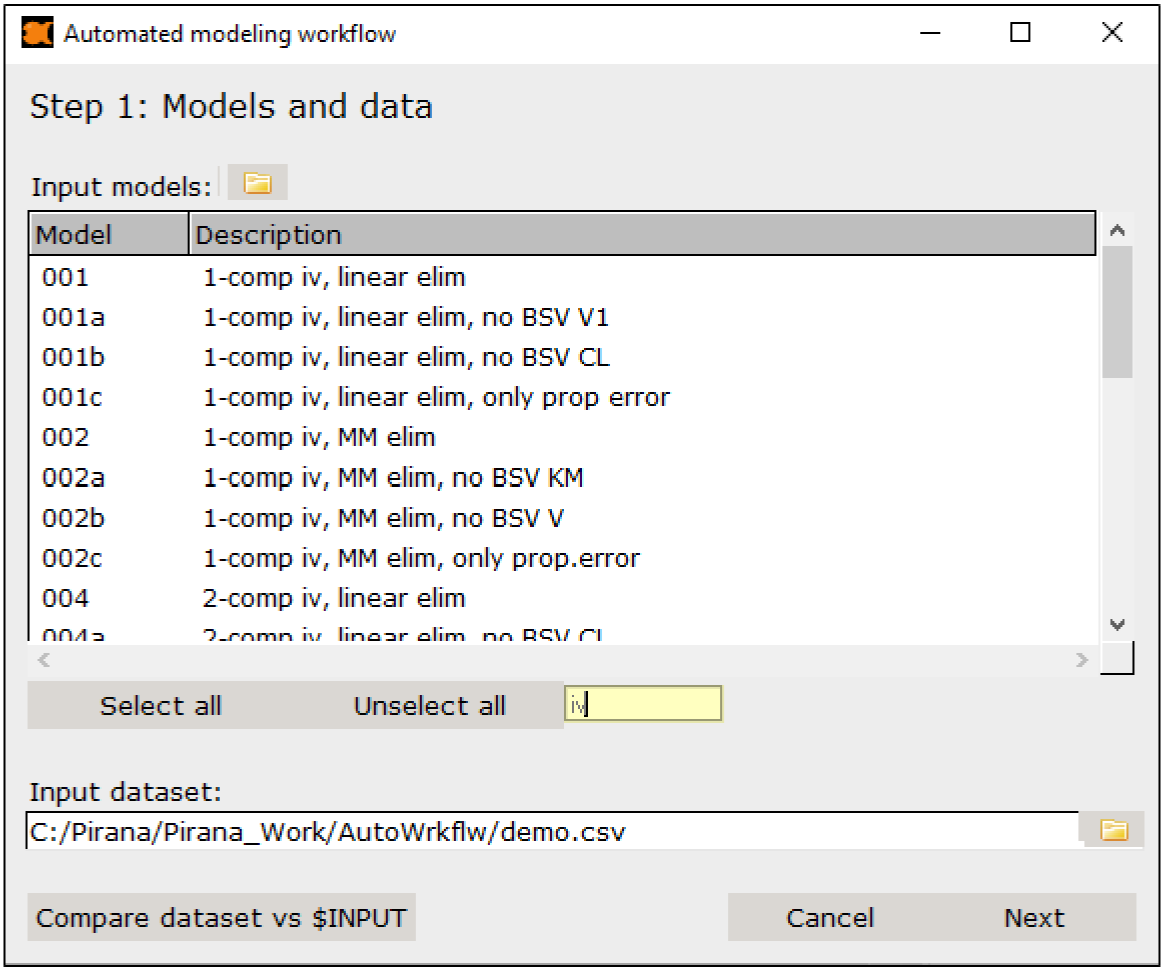
New analysis

1. Start Pirana
2. Create a new project folder somewhere on your hard-drive (or cluster).
3. Browse into this folder (with Pirana) and copy the file `demo.csv` that is included in the Pirana installation folder (`Pirana/automod_library/demo/demo.csv`).
4. In Pirana, select the **Tools > Automated modeling workflow > New analysis** menu option.

- Models and dataset
- Setting initial parameter estimates
- Folders
- PsN setup

Models and dataset

The *Step 1* dialog shows all models available in the library that can be selected for inclusion in the analysis. Use the filter field for conveniently selecting, e.g., only `iv` or only `oral` models. The dataset should of course be specified as well before advancing to the next step.



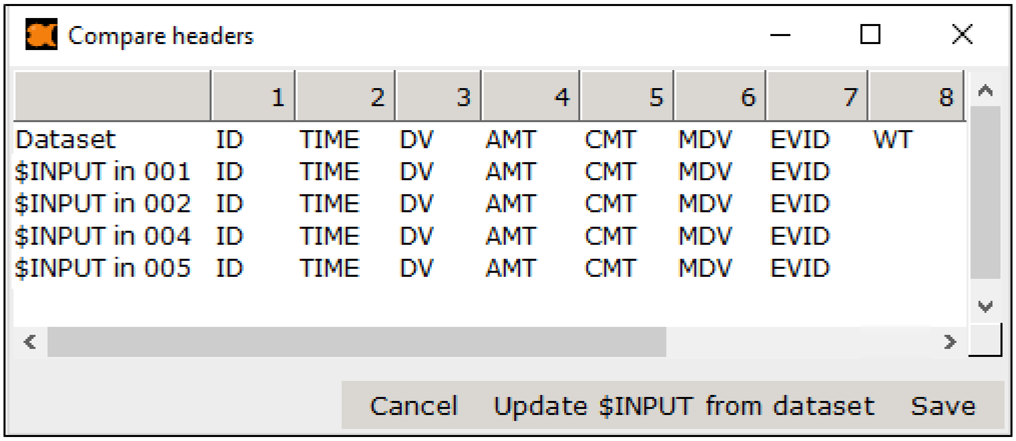
Model	Description
001	1-comp iv, linear elim
001a	1-comp iv, linear elim, no BSV V1
001b	1-comp iv, linear elim, no BSV CL
001c	1-comp iv, linear elim, only prop error
002	1-comp iv, MM elim
002a	1-comp iv, MM elim, no BSV KM
002b	1-comp iv, MM elim, no BSV V
002c	1-comp iv, MM elim, only prop.error
004	2-comp iv, linear elim
004a	2-comp iv, linear elim, no BSV CL

C:/Pirana/Pirana_Work/AutoWrkflw/demo.csv

5. For this analysis, select all iv models by entering `iv` in the filter field of the *Step 1* dialog.
6. Press the **Ctrl** key and select models **001**, **002**, **004**, and **005** in the list.

   When models and dataset have been selected, you should check whether the $INPUT record in the models matches with the headers in the dataset.

7. Click **Compare dataset vs $INPUT**. This will bring up the *Compare headers* dialog.



	1	2	3	4	5	6	7	8
Dataset	ID	TIME	DV	AMT	CMT	MDV	EVID	WT
$INPUT in 001	ID	TIME	DV	AMT	CMT	MDV	EVID	
$INPUT in 002	ID	TIME	DV	AMT	CMT	MDV	EVID	
$INPUT in 004	ID	TIME	DV	AMT	CMT	MDV	EVID	
$INPUT in 005	ID	TIME	DV	AMT	CMT	MDV	EVID	

If the $INPUT in the models (shown in rows 2-. . .) does not match up with the dataset (shown in row 1), you can click the button **Update $INPUT from dataset**. This will create a new $INPUT record for all models. After clicking **Save**, when the models will be written (in step 3 of the automated analysis), the $INPUT records in all models will be changed to the new one. It is left to the user to make sure that the variables used in the model are still included in $INPUT, as there is no extra check in place for that.

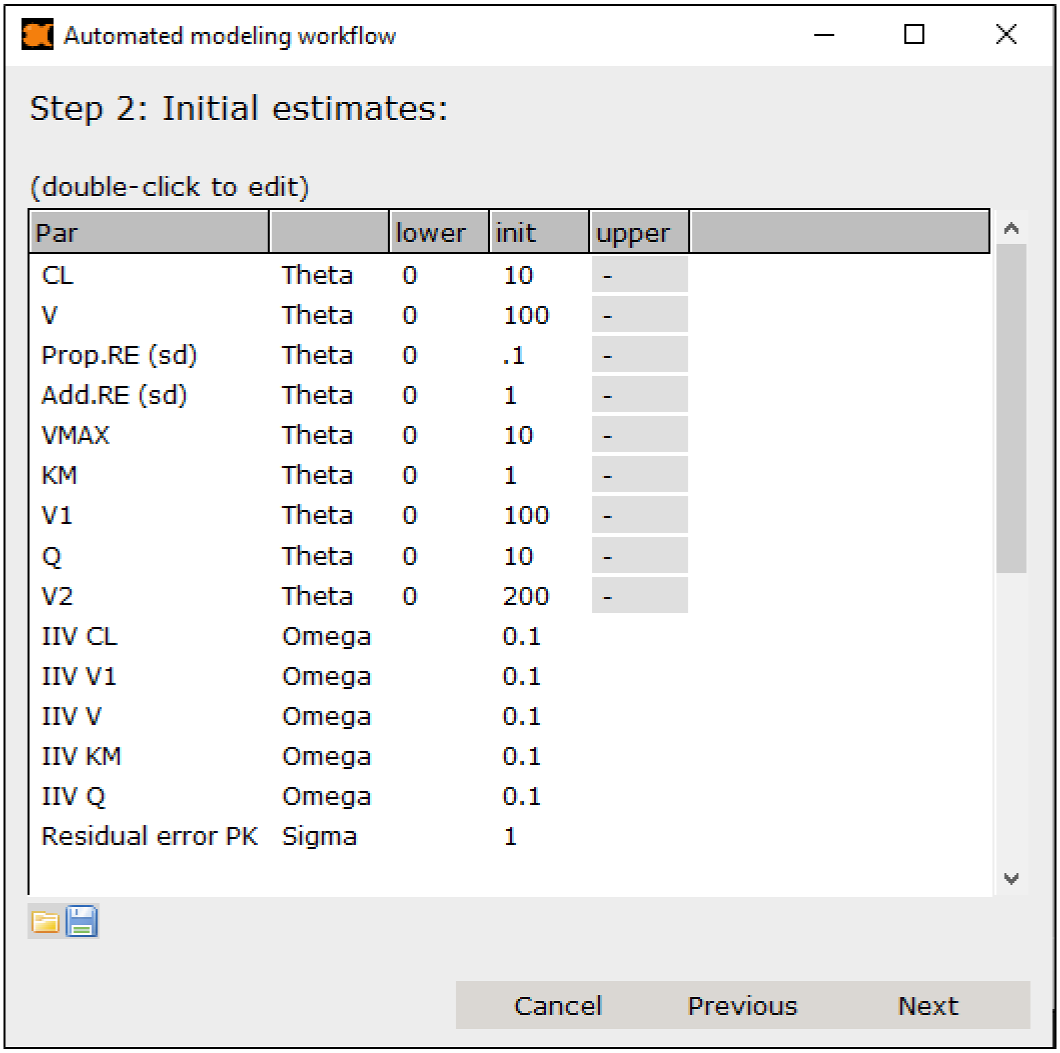
8. Click **Update $INPUT from dataset** to update the records.
9. Click **Save** and then **Next** to advance to the next step.

Setting initial parameter estimates

In the *Step 2* dialog, set initial parameter estimates, as well as lower and upper bounds. All parameters are read from the models that were selected in the *Step 1* dialog. The parameter descriptions are defined in the models as comments to $THETA, $OMEGA, and $SIGMA blocks, e.g.

```
$THETA
(0, 5, 100); CL
(0, 5, 100); V

$OMEGA
(0.1); CL
(0.1); V

$SIGMA
0.05 ; proportional error
```

Note: Correlations in $OMEGA and $SIGMA cannot be specified for an automated analysis. That is, only the diagonal elements of $OMEGA and $SIGMA can be specified in the template models if you want to update them in this step. You can still include models that have full $OMEGA or $SIGMA blocks as template model, however you cannot provide descriptions (as comments) to the parameters in the block, and you cannot update them in this step of the analysis.



Step 2: Initial estimates:

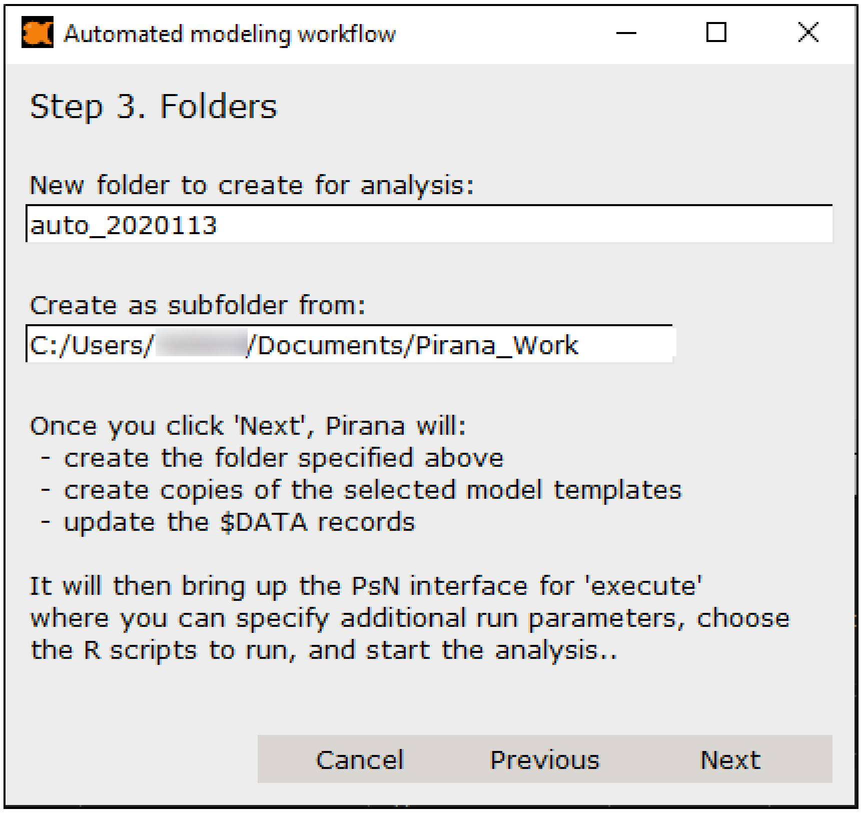
(double-click to edit)

Par		lower	init	upper	
CL	Theta	0	10	-	
V	Theta	0	100	-	
Prop.RE (sd)	Theta	0	.1	-	
Add.RE (sd)	Theta	0	1	-	
VMAX	Theta	0	10	-	
KM	Theta	0	1	-	
V1	Theta	0	100	-	
Q	Theta	0	10	-	
V2	Theta	0	200	-	
IIV CL	Omega		0.1		
IIV V1	Omega		0.1		
IIV V	Omega		0.1		
IIV KM	Omega		0.1		
IIV Q	Omega		0.1		
Residual error PK	Sigma		1		

The two icons below the parameter list can be used to save (💾) parameter definitions to a csv file and load (📂) parameters from a csv file.

1. For this analysis, leave the parameters as they are. Click **Next** to advance to the next step.

Folders



Step 3. Folders

New folder to create for analysis:

auto_2020113

Create as subfolder from:

C:/Users/[redacted]/Documents/Pirana_Work

Once you click 'Next', Pirana will:
- create the folder specified above
- create copies of the selected model templates
- update the $DATA records

It will then bring up the PsN interface for 'execute' where you can specify additional run parameters, choose the R scripts to run, and start the analysis..

Use the *Step 3* dialog to specify where to create the new models and run the analysis. By default, it will generate a new folder with a name based on the current date as a subfolder in the current folder in Pirana. This screen also lists the actions that Pirana will perform.

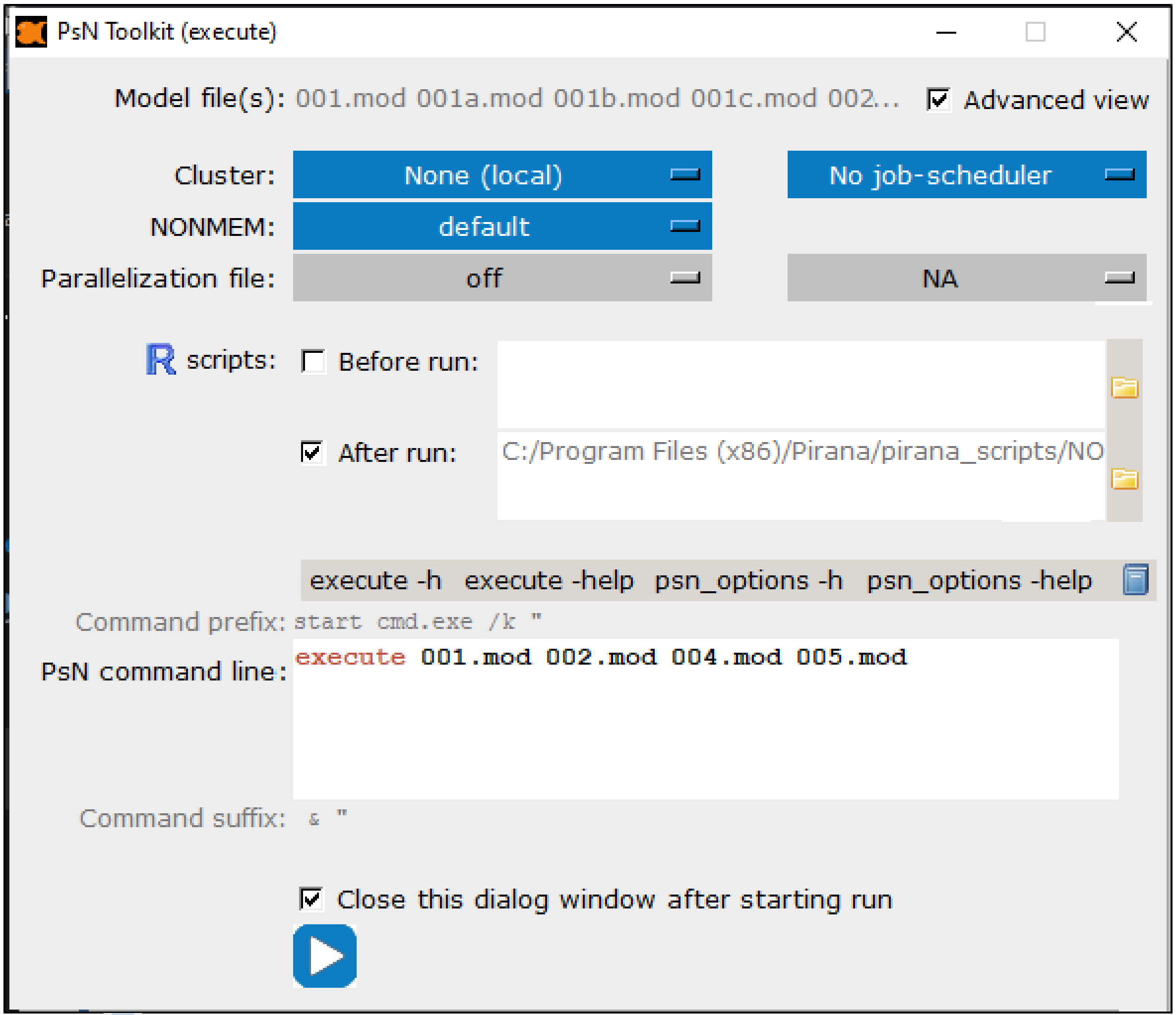
1. Use the defaults and click **Next**.

PsN setup

Pirana switches automatically to the new folder, where you will see the newly generated models. Pirana will also automatically bring up the PsN Toolkit (execute) dialog.

In this dialog, if you check the **Advanced view** box, you can select which R script(s) to run after all runs have been completed to generate goodness-of-fit plots.

2. Click 📁 next to the R scripts fields to select R scripts (or batch files) to run after (or before) the analysis step.
3. For this analysis, select the **NONMEM > Basic_GOF** plots as single document to create GOF plots for all models.



Model file(s): 001.mod 001a.mod 001b.mod 001c.mod 002... ☑ Advanced view

Cluster: None (local) No job-scheduler

NONMEM: default

Parallelization file: off NA

R scripts: □ Before run:

☑ After run: C:/Program Files (x86)/Pirana/pirana_scripts/NO

Command prefix: `start cmd.exe /k "`

PsN command line:

```
execute 001.mod 002.mod 004.mod 005.mod
```

Command suffix: `& "`

☑ Close this dialog window after starting run

4. Click ▶ to execute the workflow.

The graphical report will automatically be opened, but is also available from the **Reports** tab on the right.

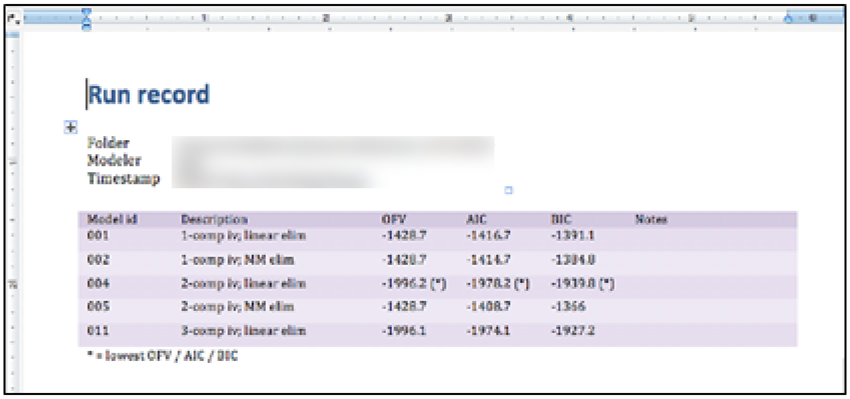
If you have not selected R scripts to be executed automatically after the analysis has completed, you can still create them afterwards by selecting the runs and running any R script from the **R** tab on the right side of the *Pirana* window.

Reports

Besides the graphical report, Pirana can generate a numeric report for the analysis, including, e.g., OFVs, basic run information and parameter estimates. This document is not generated automatically but has to be requested manually after the analysis is complete:

1. Make sure Pirana is still in the folder where the analysis was run, and then go to **Tools > Automated modeling workflow > Report**.

On the first page you will see an overview of all models included in the analysis and their respective OFV, AIC and BIC. The subsequent pages includes information on each individual run in the analysis.



Run record

Folder
Modeler
Timestamp
[redacted]

Model id	Description	OFV	AIC	BIC	Notes
001	1-comp iv; linear elim	-1428.7	-1416.7	-1391.1	
002	1-comp iv; MM elim	-1428.7	-1414.7	-1384.0	
004	2-comp iv; linear elim	-1996.2 (*)	-1978.2 (*)	-1939.8 (*)	
005	2-comp iv; MM elim	-1428.7	-1408.7	-1366	
011	3-comp iv; linear elim	-1996.1	-1974.1	-1927.2	

* = lowest OFV / AIC / BIC

Pirana and Clusters

Pirana supports interaction with Linux-based clusters on which NONMEM and/or PsN are installed. The Job-schedulers Sun Grid Engine (SGE), Torque, and Condor are supported along with SSI-type cluster managers such as MOSIX. Connecting to a cluster is established using the SSH protocol or any method that can be invoked from the command line. Two methods are available for using Pirana with a grid/cluster system, which involve installation of Pirana either on the local system or directly on the cluster server. The following paragraphs discuss these two separate methods.

Note: Single-system image clusters such as MOSIX, openMOSIX, and Kerrighed distribute processes automatically across nodes, and therefore no alternative setup is required in Pirana.

Additional information is available on the following topics:

Method 1: Server-based installation

When using this approach, Pirana is only installed on the cluster-server, not on the local machine. Pirana is executed from the local machine using ***X-over-SSH*** window tunneling. This has the advantage of requiring only one central installation of Pirana for the entire modeling group, and Pirana and other modeling software is installed in a controllable environment. A disadvantage is that the interface is usually a bit slower. Especially when using this method over larger distances (i.e., across internet), the performance of Pirana may be impaired due to the server-client transmission of the full GUI, but this of course depends on the bandwith of your connection and can be tested easily. Also, all auxiliary software (Office suite, HTML-browser, R and an R-GUI, etc.) resides on the cluster.

X-over-SSH tunneling

On the local machine it is necessary to have an X window system installed. For Linux users this is likely already installed. Mac OSX users need to install the XQuartz system. For Windows, a good X window manager is Xming, which can be obtained for free from http://sourceforge.net/projects/xming. After installation of Xming, start the Xming X window server. An alternative to Xming is Cygwin/X.

Using the cluster

If everything is set up correctly, and the X window server is started, Pirana on the cluster can be accessed through SSH, by using the SSH client. If you get an error saying that the display cannot be started on localhost, you may have to enable X window forwarding in OpenSSH or in PuTTY. When using PuTTY, it is essential to use the PuTTY terminal directly, and not `plink.exe`. The latter program can cause Pirana to crash often, probably due to terminal incompatibility. OpenSSH can also be used.

Method 2: Local installation

The other method is to install Pirana on the local machine, and connect to the cluster using Pirana and third-party SSH software. This installation approach offers a more stable interface (independent of network speed), and does not require installation of auxiliary software on the cluster. It will, however, require a few additional local installations.

First, you need to mount the cluster drive with your data on your local PC (using sshfs on Linux/Mac or ExpanDrive on Windows). For example, ExpanDrive could be used to connect to the cluster through SFTP. While ExpanDrive technically supports use on Mac, experience indicates that sshfs shows much better performance than ExpanDrive. Therefore, sshfs is recommended when using Mac. Refer to https://github.com/osxfuse/osxfuse/wiki/SSHFS. Alternatively, if a Samba server is installed on the remote cluster, a connection can be established by giving the following command:

```
NET USE Z: \ \server_name\<name> /user:<name> /persistent:yes
```

Both on Windows and Linux, the mounted remote disk space and the local location need to be specified through the NONMEM Configuration Settings page (**SSH to Cluster** sub-tab), as these settings are used by Pirana to translate local paths to paths on the remote cluster.

Secondly, an SSH client needs to be installed, which is typically already available on Linux or Mac. On Windows, PuTTY (http://www.chiark.greenend.org.uk/~sgt) and OpenSSH (download from http://sshwindows.sourceforge.net/) are good choices.

Installing public and private authentication keys

Either on Windows or Linux, type in a shell/console window: (If you use PuTTY instead of OpenSSH, use the Keypair generator program instead.)

```
ssh-keygen -t rsa
```

When asked for a passphrase, press **<Enter>**. Now a public and a private key have been created in `c:\Documents and Settings\<Name>\.ssh` (Windows) or `/home/username/.ssh` (Linux). In your home directory on the cluster, if it does not exist already, create the folder `.ssh`. In this folder, create the file `authorized_keys` (no extension) and add the contents of `id_rsa.pub` to that file and save it. Now you should be able to login without being asked for a password. If SSH asks if you want to accept the cluster as a valid host, accept. Keep your private key secret. In the Pirana NONMEM Configuration Settings page (**SSH to Cluster** sub-tab), specify the username to connect to the cluster (ssh_login).

Tip: if you experience delays (about 5 secs) when logging in to the server by SSH, this may be caused by a reverse DNS lookup. You can circumvent this by adding `useDNS no` to the file `/etc/ssh/sshd_config` on the server. Restart the ssh server for the changes to take effect: `sudo /etc/init.d/ssh restart`.

Setting up and working with a cluster over SSH on Windows

This section explains how to prepare, configure, and work with a cluster over SSH, and how to subsequently work with Pirana to execute models on the cluster.

In order to execute runs on a cluster from a local system with Pirana, SSH access to the cluster from your local computer must be available.

  - For Windows, an easy way to do this is to install PuTTY. Make sure that you install the complete version of PuTTY, including the command line tool `plink.exe`.
  - After installation, make sure Putty is available in the system path, or add the location of the PuTTY folder to the internal Pirana path via Environment variables Configuration Settings page **PATH at startup of Pirana** option.
  - On Linux and Mac OSX, `ssh` is most likely already installed.

- Any runs which are to be submitted to the cluster should be in a location on the drive that you specified as the remote cluster mount location.
- A model can be run on the cluster via either nmfe or PsN, which are described separately below.

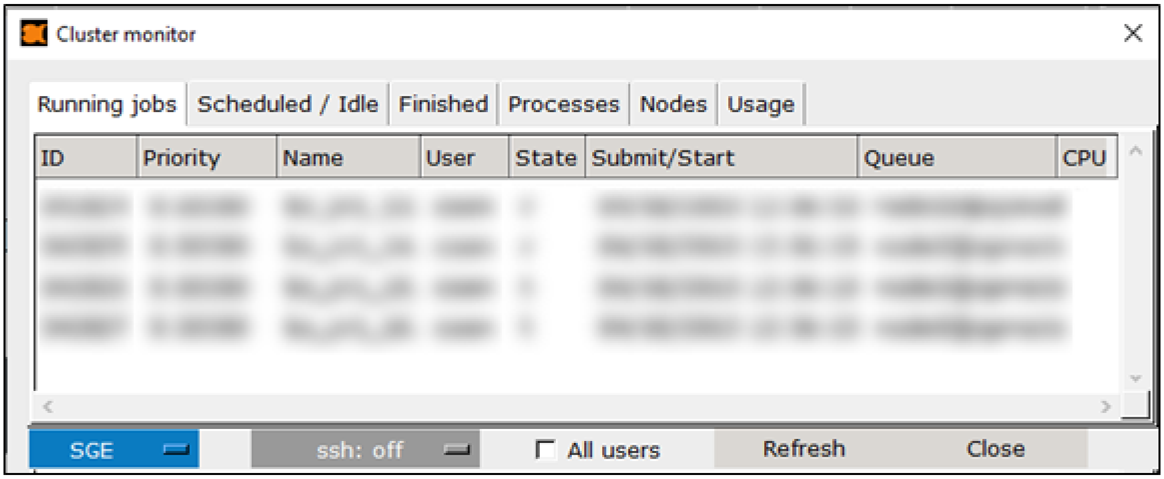
Monitoring jobs on SGE, Torque, or Condor clusters

Submitting the execution of a NONMEM model using nmfe to the SGE, Torque, or Condor can be done by selecting the **Run on SGE or Torque** from the *nmfe* or *PsN* dialogs. This submits the model using `qsub` instead of starting it directly.

If SGE, Torque, or Condor is used as the job management system, the queue may be monitored using the integrated monitor in Pirana.

1. Select **View > Cluster monitor** in the main menu
   Or
   Click the in the toolbar.
2. In the *Cluster monitor* dialog, select a tab to view different types of cluster information.
   - **Running**: Lists jobs that are currently running.
   - **Scheduled/Idle**: Lists jobs that are scheduled or are idle.
   - **Finished**: Lists jobs that have recently finished.
   - **Nodes**: Displays information about available nodes in the cluster.
   - **Usage**: Displays information about cluster usage.
3. Right-click a job to access more information about it or to kill it.

Note: All compute jobs are listed, not just NONMEM jobs.



4. Use the pull-down at the bottom of the dialog to switch the cluster being monitored.
5. Use the **SSH** pull-down to toggle use of SSH protocol.
6. Check the **All users** box to see all jobs that have been submitted by all users.
   If unchecked, only your jobs will be listed.
7. Click **Refresh** to reload the information in the dialog.

Configuration Settings

Configuring Pirana is done through the *Settings* dialog, accessed using the **File > Settings** menu option. Categories of configuration settings include:

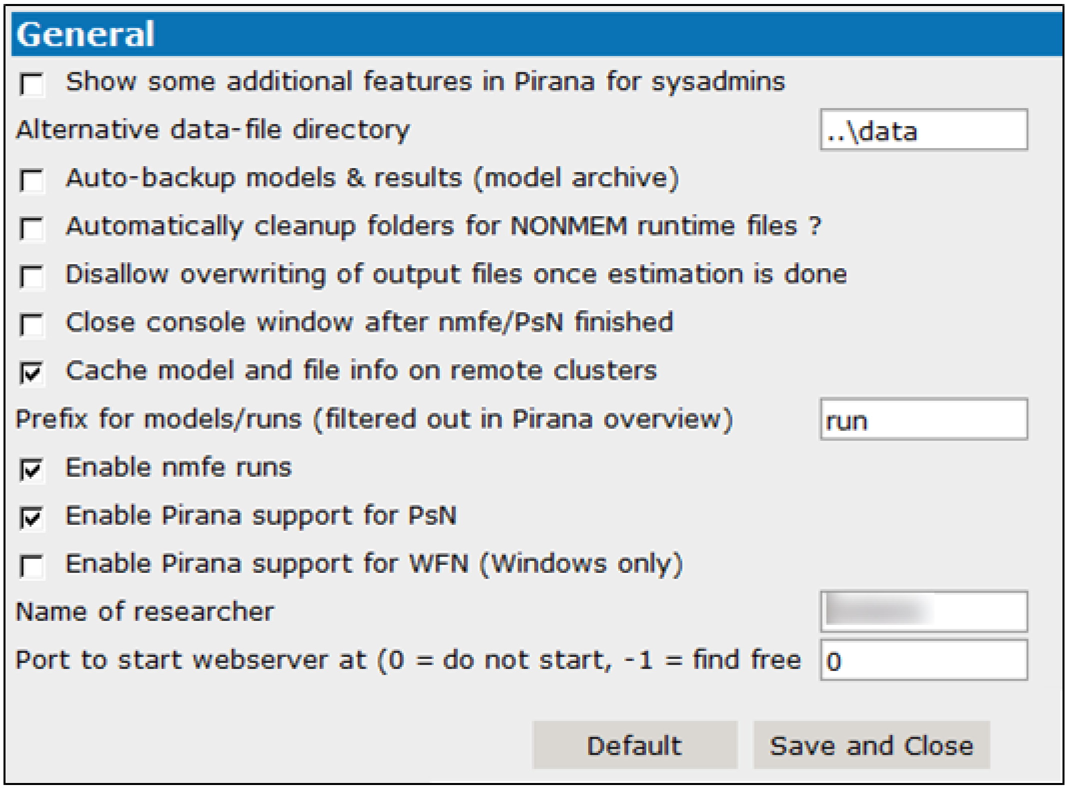
- General settings
- Appearance settings
- Miscellaneous settings
- NONMEM
- NLME
- Software integration
- Environment variables
- Run reports settings
- LaTeX settings

It is important to press **Save and Close** to save any changes made to settings on the current panel or tab as the defaults and close the *Settings* dialog. Press **Default** to return the settings on that panel or tab to their values at the last save action.

Note: Once modifications are made and you have saved them, restart Pirana for the changes to take effect.

General settings

1. Select **File > Settings**.
2. Select **General** in the left list.



General

- ☐ Show some additional features in Pirana for sysadmins
- Alternative data-file directory: ..\data
- ☐ Auto-backup models & results (model archive)
- ☐ Automatically cleanup folders for NONMEM runtime files ?
- ☐ Disallow overwriting of output files once estimation is done
- ☐ Close console window after nmfe/PsN finished
- ☑ Cache model and file info on remote clusters
- Prefix for models/runs (filtered out in Pirana overview): run
- ☑ Enable nmfe runs
- ☑ Enable Pirana support for PsN
- ☐ Enable Pirana support for WFN (Windows only)
- Name of researcher: [redacted]
- Port to start webserver at (0 = do not start, -1 = find free: 0

Most of these settings do not need altering to work with Pirana.

Show some additional features in Pirana for sysadmins: When checked and Pirana is restarted, an additional icon (📒) appears in the toolbar, allowing system administrators to view the command execution logs for all users. The date/time, username, project name, model name, and command executed are listed and can be exported as a csv file or the log can be deleted.

Auto-backup models & results (model archive): When checked, models and results are copied to a backup folder. (Each folder containing models has a sub-folder for backup purposes.

Automatically cleanup folders for NONMEM runtime files?: When checked, runtime files are removed after run completes.

Disallow overwriting of output files once estimation is done: When unchecked, you can re-execute the same run multiple times and Pirana will create a new subdirectory, sequentially numbered, for each run's outputs. It will also copy the main results to the root folder directory, which makes it (the most recent results) readily available. When checked, you can only execute the run once and need to create a new run in order to re-execute.

Close console window after nmfe/PsN finished: When checked, the console window is closed automatically after nmfe/PsN run completes.

Cache model and file info on remote clusters: When checked, colors and model/folder information located on clusters (Remote Mount Location) are saved into a local cache.

Prefix for models/runs (filtered out in Pirana overview): Enter a prefix to prepend to any model or run file names.

Enable nmfe runs: When checked, NONMEM models can be run directly using nmfe.

Enable Pirana support for PsN: When checked, NONMEM models can be run by Pirana through PsN.

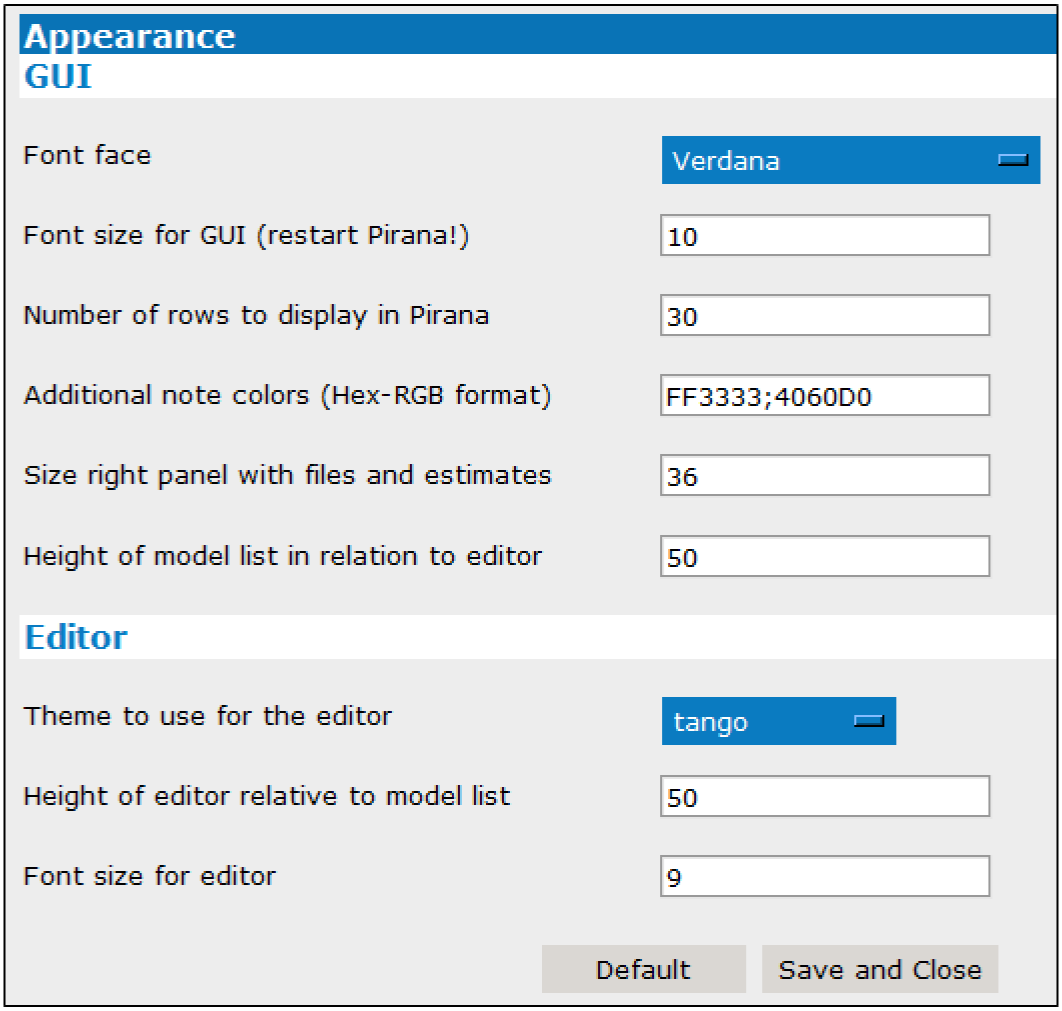
Name of researcher: Your name.

Port to start webserver at: Enter the port for the Pirana webserver. Enter `0` to disable the Pirana webserver, `-1` to automatically choose any available port, or any valid free port number.

Note: Once modifications are made and you have saved them, restart Pirana for the changes to take effect.

Appearance settings

1. Select **File > Settings**.
2. Select **Appearance** in the left list.



Appearance

GUI

Font face: Verdana

Font size for GUI (restart Pirana!): 10

Number of rows to display in Pirana: 30

Additional note colors (Hex-RGB format): FF3333;4060D0

Size right panel with files and estimates: 36

Height of model list in relation to editor: 50

Editor

Theme to use for the editor: tango

Height of editor relative to model list: 50

Font size for editor: 9

GUI preferences

Font face: Select the font for text displayed in the interface.

Font size for GUI: Enter the font size for text displayed in Pirana.

Number of rows to display in Pirana: Set the number of rows that are visible in table.

Additional note colors: User-defined colors, in addition to default colors, that are used to mark/filter models. These appear in the list when ▦ is selected in the main toolbar. To add additional colors to the list, go to https://www.rapidtables.com/convert/color/rgb-to-hex.html, use the sliders to get a color you like, click **Convert**, and you will get a 6 digit hex code (e.g., #9425EF, which is deep purple). Enter the code (omitting the #), e.g., 9425EF, into the field, and the color will appear as an additional option in the list.

Editor preferences

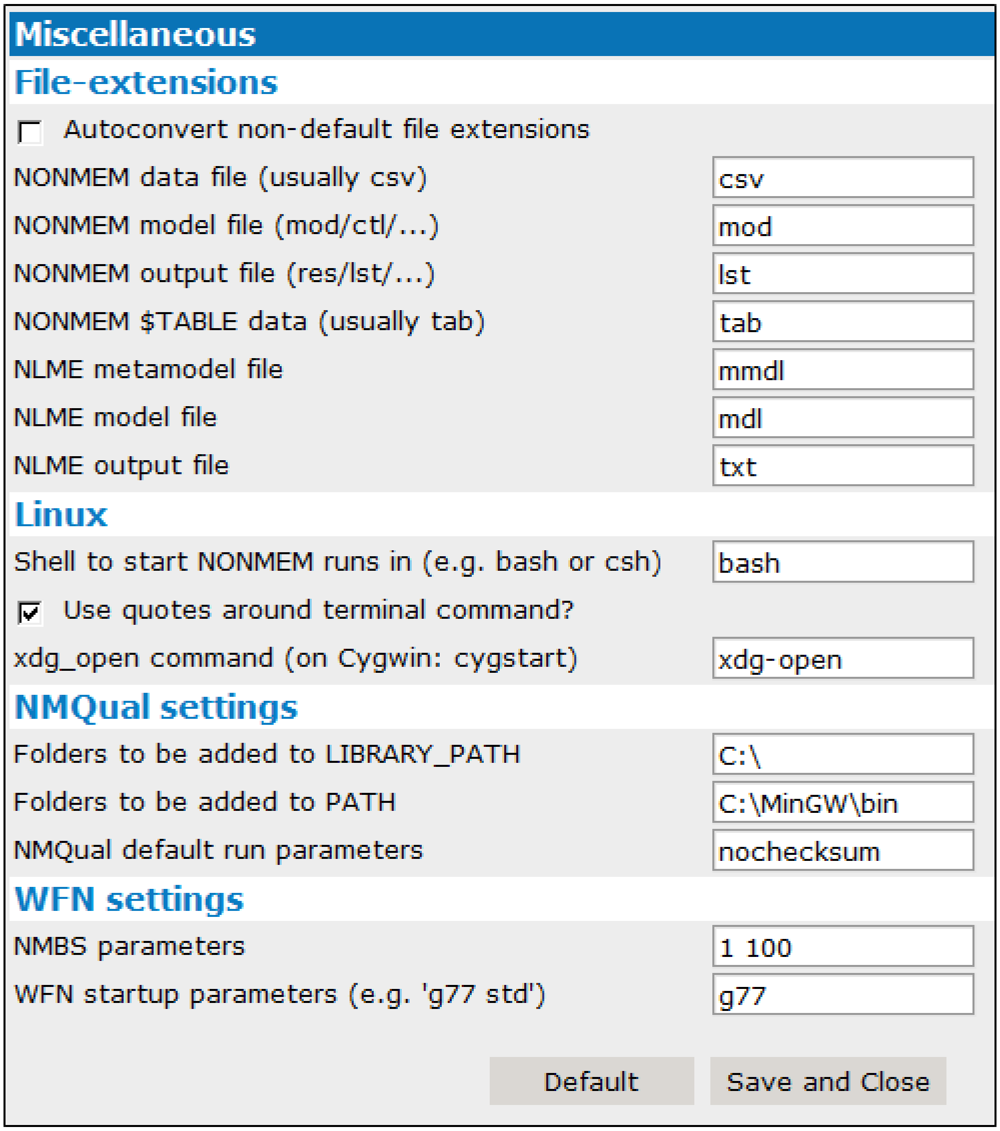
Theme to use for the editor: Color theme for the built-in editor.

Font size for editor: Enter the font size for text displayed in the editor.

Note: Once modifications are made and you have saved them, restart Pirana for the changes to take effect.

Miscellaneous settings

1. Select **File > Settings**.
2. Select **Miscellaneous** in the left list.



Miscellaneous

File-extensions

□ Autoconvert non-default file extensions

NONMEM data file (usually csv)	csv
NONMEM model file (mod/ctl/...)	mod
NONMEM output file (res/lst/...)	lst
NONMEM $TABLE data (usually tab)	tab
NLME metamodel file	mmdl
NLME model file	mdl
NLME output file	txt

Linux

Shell to start NONMEM runs in (e.g. bash or csh)	bash

☑ Use quotes around terminal command?

xdg_open command (on Cygwin: cygstart)	xdg-open

NMQual settings

Folders to be added to LIBRARY_PATH	C:\
Folders to be added to PATH	C:\MinGW\bin
NMQual default run parameters	nochecksum

WFN settings

NMBS parameters	1 100
WFN startup parameters (e.g. 'g77 std')	g77

File Extensions

Autoconvert non-default file extensions: For NONMEM models only. When checked, if there are any .ctl or .mod files in the current directory, Pirana will propose to rename those files (i.e., change their extension to the default **NONMEM model file** setting).

Adjust the file extensions to associate with each of the following items/applications. (It is recommended that these are not changed.)

NONMEM data file **NONMEM model file** **NONMEM output file** **NONMEM $TABLE data**

NLME metamodel file **NLME model file** **NLME output file**

Linux

Shell to start NONMEM runs in: Enter the type of Linux shell in which to run NONMEM.

Use quotes around terminal command: Check this box to add quotes around commands echoed to the terminal.

xdg_open command: Enter the command to use to open NONMEM.

NMQual Settings

Folders to be added to LIBRARY_PATH: Refer to the NMQual set up instructions.

Folders to be added to PATH: Refer to the NMQual set up instructions.

NMQUAL default run parameters: Enter any parameters to use to setup the run environment.

WFN Settings

NMBS parameters: Enter any NMBS command arguments to use to execute bootstrap NONMEM runs.

WFN startup parameters: Enter any parameters to use to setup the run environment.

Note: Once modifications are made and you have saved them, restart Pirana for the changes to take effect.

NONMEM

Existing NONMEM installations can be added to Pirana via the *Settings* window, in the NONMEM section. Here, local, remote, and cluster installations can be added for use in Pirana.

Note: Setting up NONMEM is not necessary if PsN is used.

The following topics are discussed in this section:

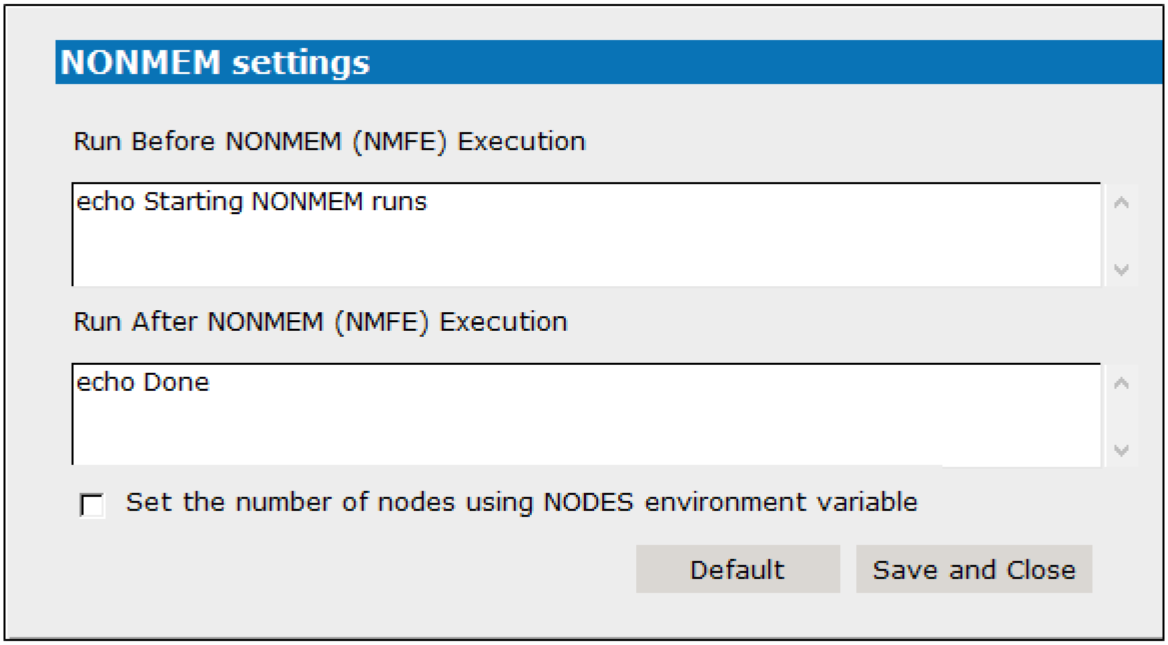
- Define commands to execute before or after NONMEM
- Set up profile for local installation
- Set up profile for remote installation
- Set up profile for NONMEM cluster
- Edit profile of NONMEM cluster

Note: Once modifications are made and you have saved them, restart Pirana for the changes to take effect.

Define commands to execute before or after NONMEM

If there are any commands that need to be executed before starting the NONMEM run, enter them in the NONMEM tab.

1. Select **File > Settings**.
2. Select **NONMEM** in the left list.



NONMEM settings

Run Before NONMEM (NMFE) Execution

```
echo Starting NONMEM runs
```

Run After NONMEM (NMFE) Execution

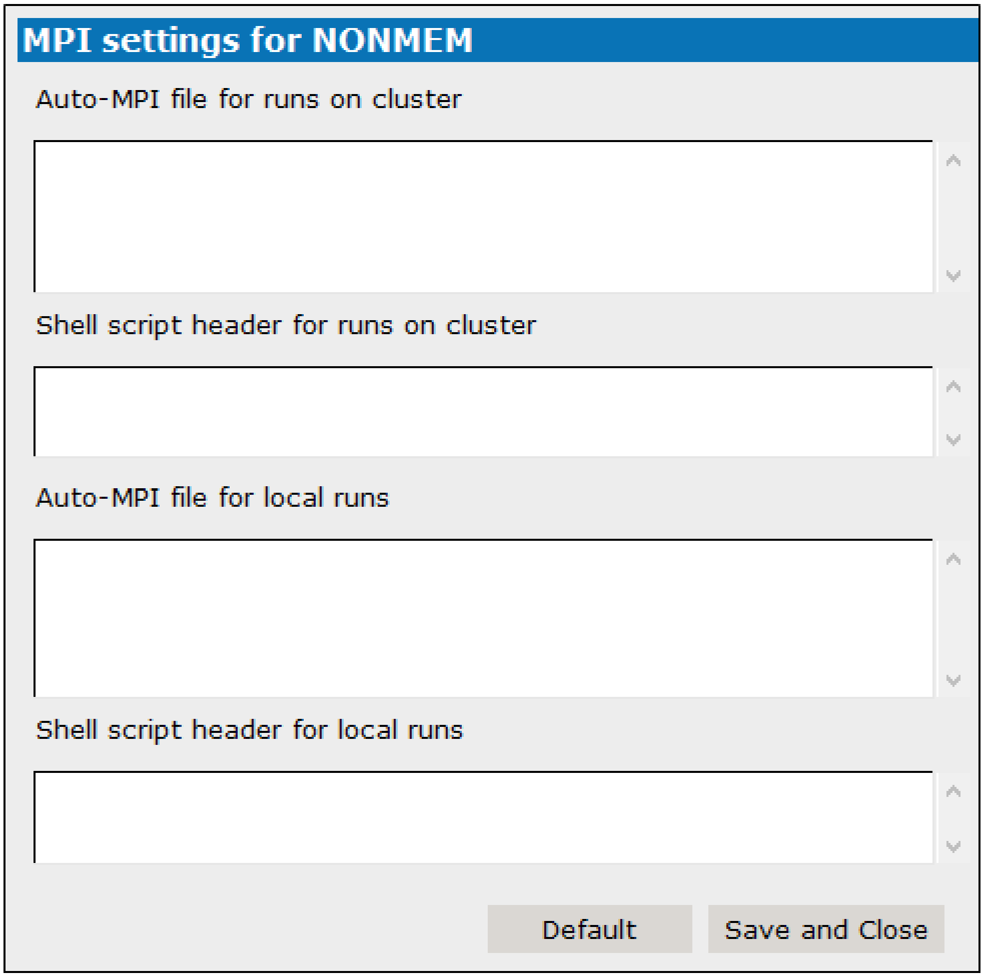
```
echo Done
```

□ Set the number of nodes using NODES environment variable

3. Enter any commands to execute prior to the NONMEM run in the first field.
4. Enter any commands to execute after the NONMEM run in the second field.
5. Check the **Set the number of nodes**... box to determine the number of nodes by looking at the NODES environment variable setting.

MPI scripts settings for parallelization

1. Select **File > Settings**.
2. In the dialog, select **MPI settings** (under NONMEM) from the list on the left.



MPI settings for NONMEM

Auto-MPI file for runs on cluster

Shell script header for runs on cluster

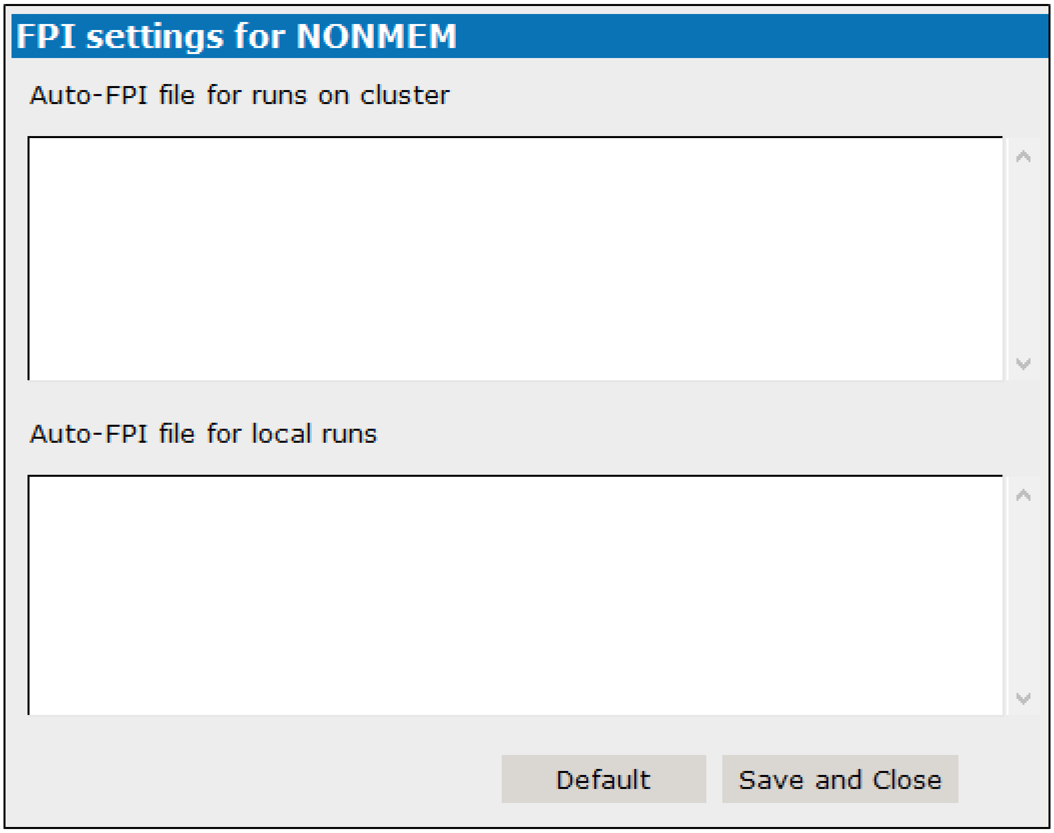
Auto-MPI file for local runs

Shell script header for local runs

3. For cluster runs, edit the script saved to the Auto-MPI file in the **Auto MPI file Clusters** field.
4. Enter the **Shell Script Header** to include in the Auto-MPI file.
5. For local runs, edit the script saved to the Auto-MPI file in the **Auto MPI file Local** field.
6. Enter the **Shell Script Header** for the Auto-MPI file.

FPI scripts settings for parallelization

1. Select **File > Settings**.
2. In the dialog, select **FPI settings** (under NONMEM) from the list on the left.

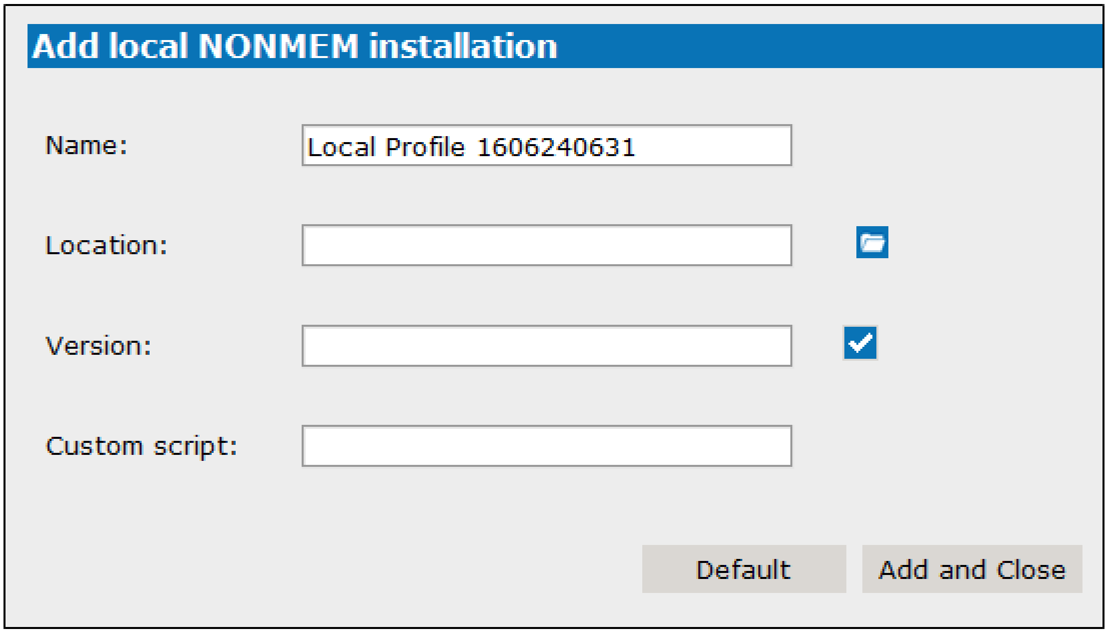


FPI settings for NONMEM
Auto-FPI file for runs on cluster
Auto-FPI file for local runs
Default
Save and Close

3. For cluster runs, edit the script saved to the Auto-FPI file in the **Auto-FPI file for runs on cluster** field.
4. For local runs, edit the script saved to the Auto-FPI file in the **Auto-FPI file for local runs** field.

Set up profile for local installation

1. Select **File > Settings**.
2. Select **Local Profiles** (under **NONMEM**) in the left list.
3. Edit the name for the new profile in the **Name** field, if desired.



Add local NONMEM installation

Name: Local Profile 1606240631

Location: 📂

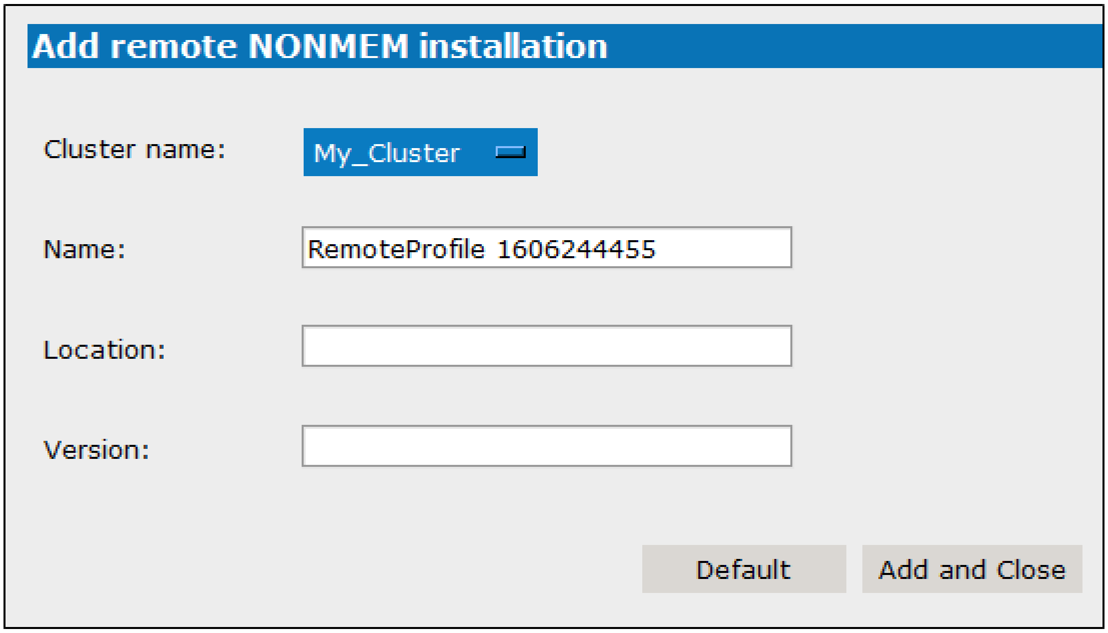
Version: ☑

Custom script:

4. Enter the full path to the **Location** of the NONMEM installation (click 📂 to use the file browser).
5. Enter the NONMEM **Version** (click ☑ to have Pirana identify the version).
6. In the **Custom script** field, list any arguments/keywords to use when starting up the environment.

Set up profile for remote installation

1. Select **File > Settings**.
2. Select **Remote Profiles** (under **NONMEM**) in the left list.



Add remote NONMEM installation

Cluster name: My_Cluster

Name: RemoteProfile 1606244455

Location:

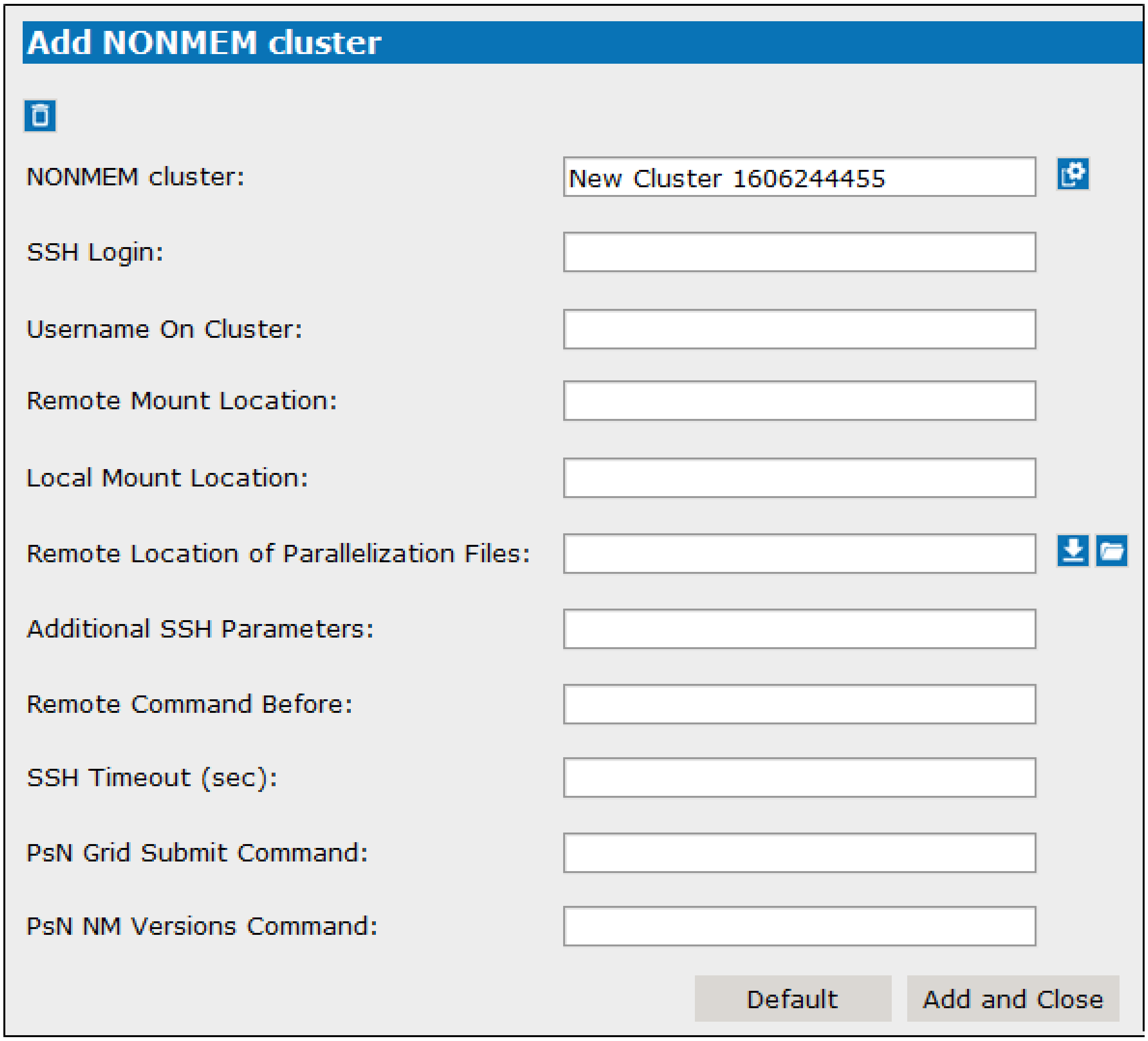
Version:

3. Edit the name for the new profile in the **Name** field, if desired.
4. Select the cluster on which the remote installation resides from the **Cluster name** pulldown.
5. Enter the full path to the **Location** of the NONMEM installation.
6. Enter the NONMEM **Version**.

Set up profile for NONMEM cluster

If the remote cluster folder can be mounted as a drive letter on your system, and you have installed PuTTy (http://www.chiark.greenend.org.uk/~sgtatham/putty/), cluster access may now be configured in Pirana.

1. Select **File > Settings**.
2. Select **Clusters** (under **NONMEM**) in the left list.



Add NONMEM cluster

NONMEM cluster:	New Cluster 1606244455
SSH Login:	
Username On Cluster:	
Remote Mount Location:	
Local Mount Location:	
Remote Location of Parallelization Files:	
Additional SSH Parameters:	
Remote Command Before:	
SSH Timeout (sec):	
PsN Grid Submit Command:	
PsN NM Versions Command:	

3. Enter a name for the **NONMEM cluster** in the field.

   Click ⚙ to have Pirana locate the ssh-keygen.

4. In the field **SSH Login** field, enter/edit the command for connecting to the cluster.

   For example: `ssh user@server.domain.ext` (e.g., plink -l smith -pw xxx 10.181.0.1)

   If PuTTY is used, this command will start with `plink`, followed by the username and password for the cluster, and the name or IP address of the cluster access node.

   Pirana needs passwordless SSH-access to the cluster, so make sure that an RSA key pair is installed (explained in the “Installing public and private authentication keys” section). If you use

PuTTY on Windows, you can also choose to supply the password on the command line instead as follows:

```
plink -l username -pw password server.domain.ext
```

5. In the **Username On Cluster** field, enter/edit the username you use to log into the cluster. (e.g., smith)
6. In the **Remote Mount Location** field, enter/edit the name a folder on the cluster that you have mounted as a local drive.

   For example: `/home/user` (e.g., /home/smith)
7. In the **Local Mount Location** field, enter/edit the drive-letter on the local system that corresponds with the remote cluster path defined in the previous field.

   For example: `X:` (e.g., V:)
8. In the **Remote Location of Parallelization Files** field, enter/edit the remote machine path to the folder where the parallelization files are stored.

   Click ⬇ to download the folder from the remote machine to a local folder.

   Click 📂 to navigate to and select the folder.
   (e.g., /home/smith/parfiles)
9. If there are any SSH parameters that need to be set, enter/edit the information in the **Additional SSH Parameters** field.
10. If there are any commands that need to be executed before connecting to the remote machine, enter/edit them in the **Remote Command Before** field.
    (e.g., source /etc/profile)
11. Enter/Edit the amount of time (in seconds), in the **SSH Timeout** field, that SSH should wait for a response from the cluster before returning a timed out error. (e.g., 8)
12. To use PsN to submit a NONMEM job to the grid, enter/edit the PsN command to execute a job on the grid in the **PsN Grid Submit Command** field.
13. Enter/Edit the PsN command for executing the NONMEM job in the **PsN NM Versions Command** field.

Once **Add and Close** is pressed, the cluster name appears under **NONMEM** > **Clusters** in the tree on the left. Selecting the cluster name in the tree displays a series of tabs for editing and further defining of the cluster. (See “Edit profile of NONMEM cluster”.)

Clicking 🗑 deletes the cluster being defined in the *Add NONMEM cluster* panel and closes the *Settings* dialog.

Edit profile of NONMEM cluster

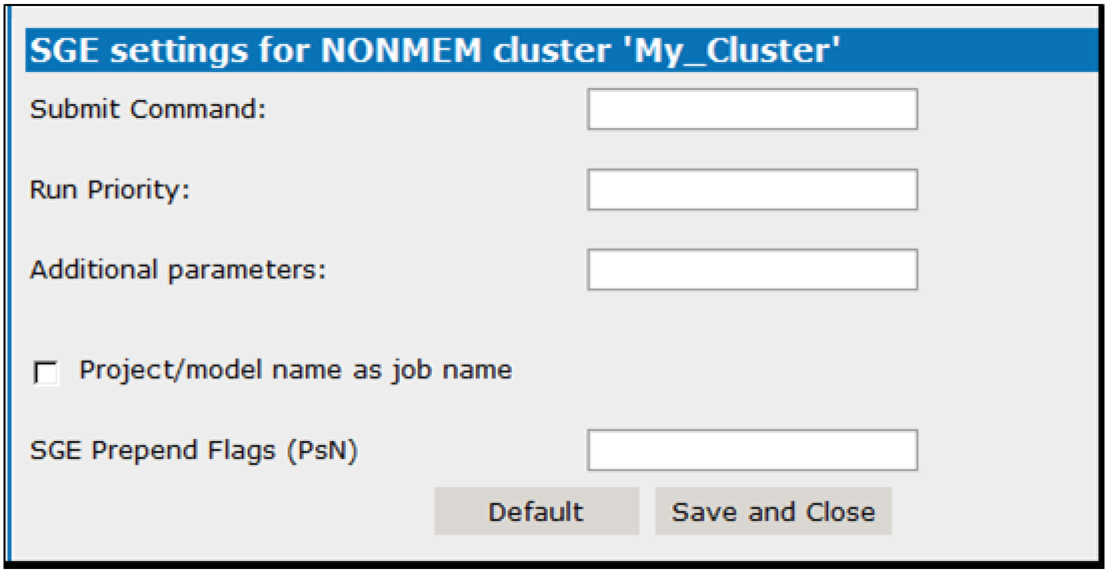
1. Select **File > Settings**.
2. In the dialog, select the name of the NONMEM cluster from the list on the left.

The SSH to Cluster tab is presented showing the settings for the cluster loaded. (See “Set up profile for NONMEM cluster” for definitions of the options.)

3. Modify the options as needed or select one of the other tabs:
   SGE cluster settings
   Torque cluster settings
   Condor cluster settings
   MPI scripts settings for parallelization
   FPI scripts settings for parallelization

SGE cluster settings

1. Select **File > Settings**.
2. In the dialog, select the name of the NONMEM cluster from the list on the left.
3. Select the **SGE** tab.



SGE settings for NONMEM cluster 'My_Cluster'

Submit Command:

Run Priority:

Additional parameters:

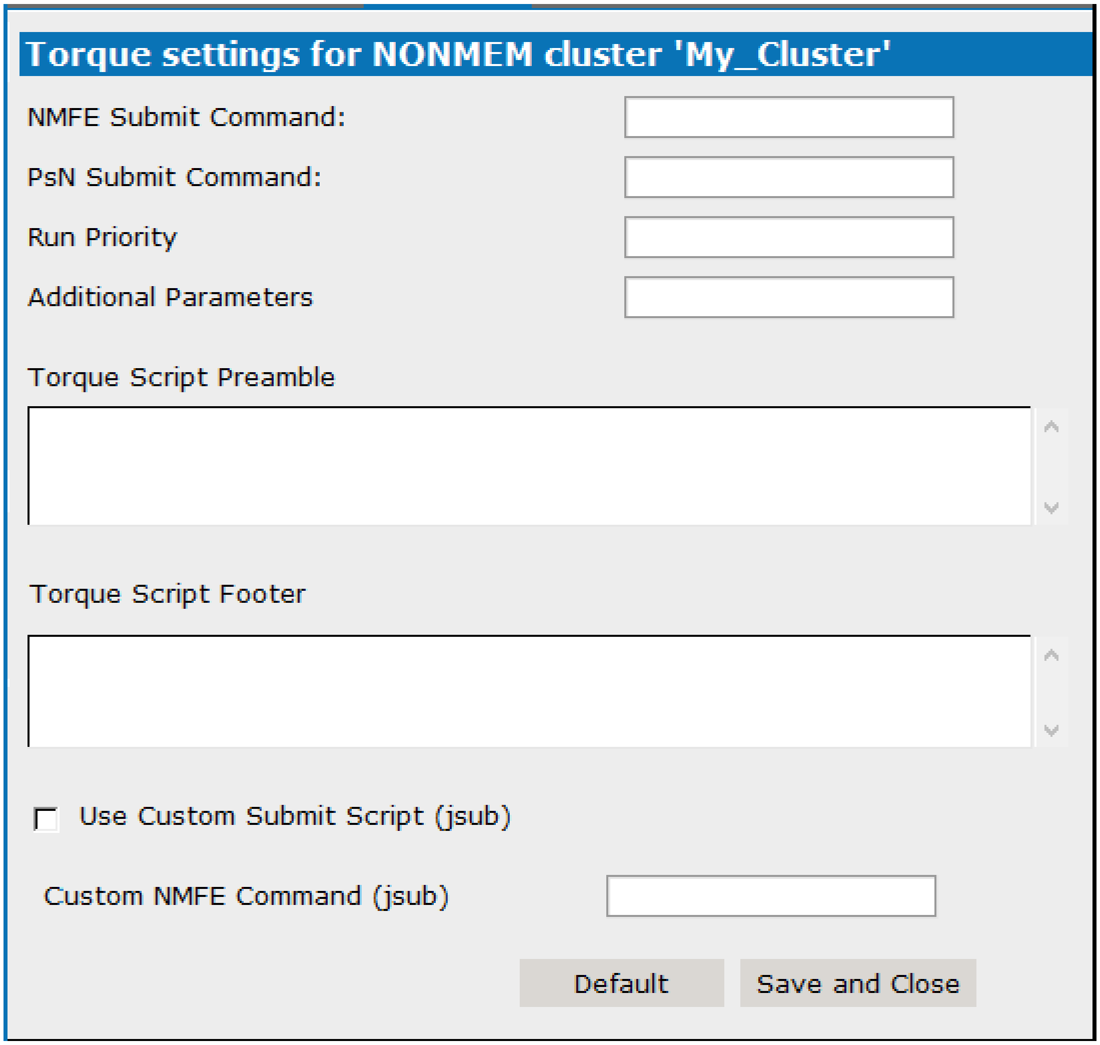
□ Project/model name as job name

SGE Prepend Flags (PsN)

4. Enter/Edit the command to use when submitting a job to SGE in the **Submit Command** field.
5. Enter/Edit the priority (integer) of jobs submitted to the SGE cluster in the **Run Priority** field.
6. Enter/Edit other `qsub` options to include in the submission command in the **Additional Parameters** field.
7. Check the **Project/model-name as job-name** box to use the project/model-name as the name for the job.
8. When using PsN, use the **SGE Prepend Flags** field to specify any SGE flags to submit to PsN.

Torque cluster settings

1. Select **File > Settings**.
2. In the dialog, select the name of the NONMEM cluster from the list on the left.
3. Select the **Torque** tab.



4. If using NMFE, enter/edit the command to use when submitting a job to Torque in the **NMFE Submit Command** field.
5. If using PsN, enter/edit the PsN command to use when submitting a job to Torque in the **PsN Submit Command** field.
6. Enter/Edit the priority (integer) of jobs submitted to the Torque cluster in the **Run Priority** field.
7. Enter/Edit any other options to include in the submission command in the **Additional Parameters** field.
8. Enter/Edit any directives to be done prior to job submission in the **Torque Script Preamble** field.

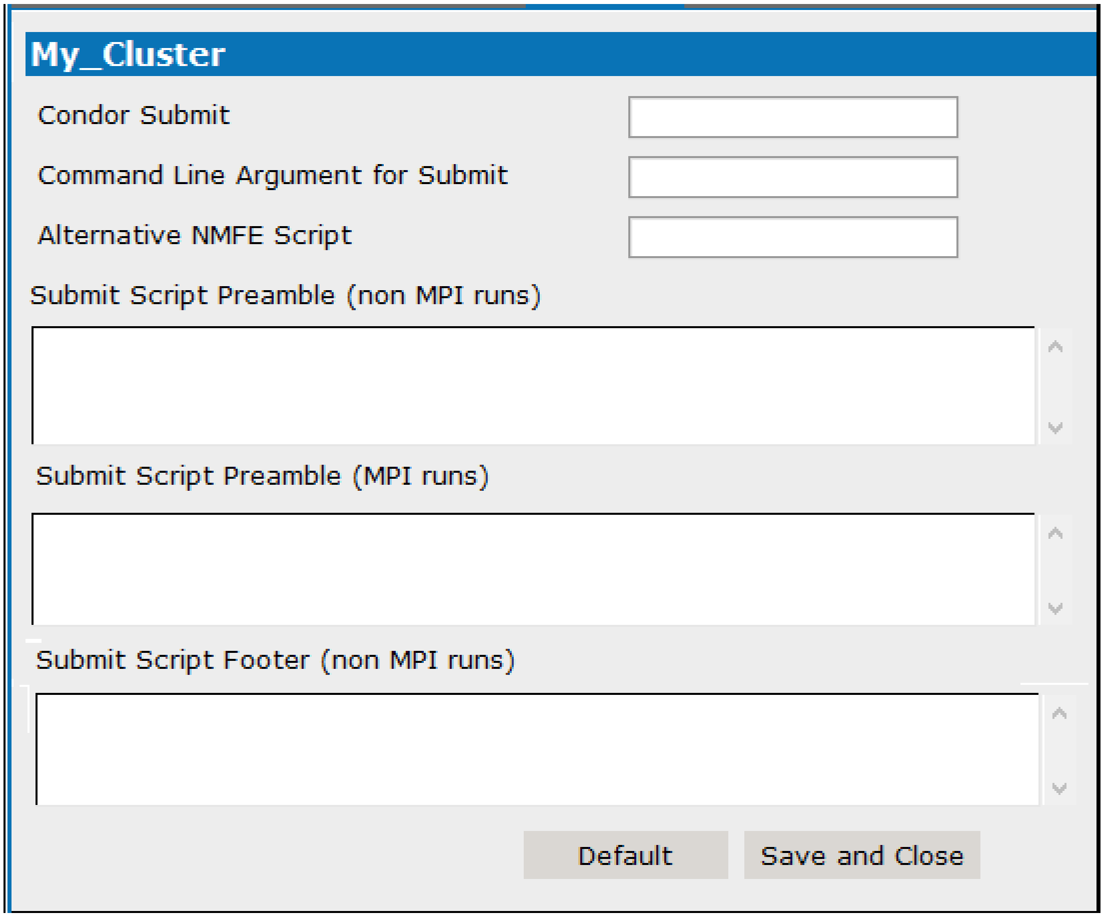
   Information requested in this field will appear at the beginning of the job’s output file.
9. Enter/Edit any directives to be done after job completion in the **Torque Script Footer** field.

   Information requested in this field will appear at the end of the job’s output file.

10. Check the **Use Custom Submit Script** box and enter the associated NMFE command in the **Custom NMFE Command** field.

Condor cluster settings

1. Select **File > Settings**.
2. In the dialog, select the name of the NONMEM cluster from the list on the left.
3. Select the **Condor** tab.


My_Cluster
Condor Submit
Command Line Argument for Submit
Alternative NMFE Script
Submit Script Preamble (non MPI runs)
Submit Script Preamble (MPI runs)
Submit Script Footer (non MPI runs)
Default
Save and Close

4. Enter/Edit the command to use when submitting a job to Condor in the **Condor Submit** field.
5. Enter/Edit any arguments to include in the submit command in the **Command Line Argument for Submit** field.
6. Enter/Edit the command to use when submitting a job via NMFE.
7. Enter/Edit any directives to be done prior to job submission in the **Submit Script Preamble** fields, the first field is for **non-MPI** runs, the second field is for **MPI** runs.

   Information requested in this field will appear at the beginning of the job’s output file.
8. Enter/Edit any directives to be done after job completion in the **Submit Script Footer** field.

   Information requested in this field will appear at the end of the job’s output file.

NLME

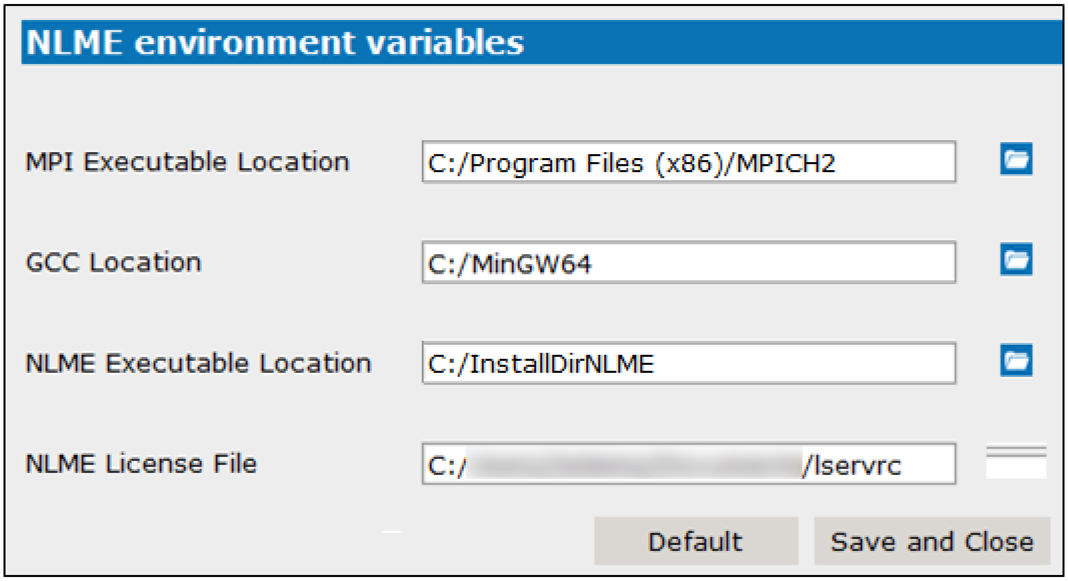
The following topics are discussed in this section:

- Set NLME environment variables
- Set up profile for NLME
- Edit an NLME profile

Note: Once modifications are made and you have saved them, restart Pirana for the changes to take effect.

Set NLME environment variables

1. Select **File > Settings**.
2. Select **NLME** in the left list.



MPI Executable Location: C:/Program Files (x86)/MPICH2

GCC Location: C:/MinGW64

NLME Executable Location: C:/InstallDirNLME

NLME License File: C:/[redacted]/lservrc

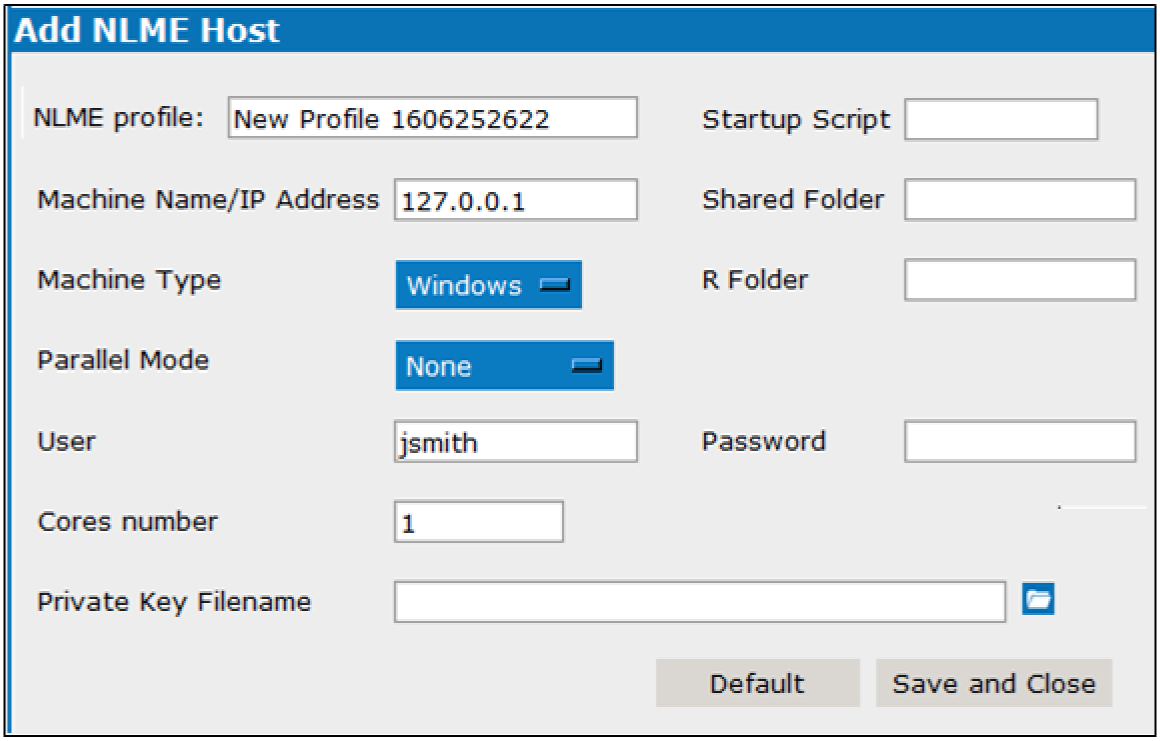
3. Enter the path to the folder where the **MPI Executable** is installed in the field (e.g., `C:/Program Files (x86)/MPICH2`).
4. Enter the path to the folder where the GCC component is installed in the **GCC Location** field (e.g., `C:/MinGW64`).
5. Enter the path to the **NLME Executables Location** in the field.
6. Enter the path to the **NLME License File Location** in the field.

Note: If the license file is provided in the same directory as the NLME executables, this field can be left blank.

Click 📂 to use the browser to locate the folders.

Set up profile for NLME

1. Select **File > Settings**.
2. Select **Profiles** (under **NLME**) in the left list.



Add NLME Host

NLME profile: New Profile 1606252622

Startup Script

Machine Name/IP Address 127.0.0.1

Shared Folder

Machine Type Windows

R Folder

Parallel Mode None

User jsmith

Password

Cores number 1

Private Key Filename

3. Edit the name in the **Profile Name** field, if desired.
4. In the **Startup Script** field, enter the script to execute on the remote host to setup the run environment.
5. Enter the machine name or the IP address where the execution will be performed in the **Machine Name/IP Address** field.
6. Enter the location where the application can write results/temporary files on the remote machine in the **Shared Folder** field.
7. Select the **Machine Type** from the pull-down. Choose from **Windows** or **Linux**
8. Specify the path to the R program on the remote machine in the **R Folder** field.
9. Select the parallelization mode to use from the **Parallel Mode** pull-down.

   Windows: None, MPI, LSF, Multicore
   Linux: None, MPI, LSF, Multicore, Torque, SGE, LSF_MPI, Torque_MPI, SGE_MPI
10. In the **User** and **Password** fields, enter the login credentials for the host. These are required to use the grid, unless using a private key file.(The password is always required when running on Linux.)
11. Enter the number of computational cores available on the grid or for any parallelization in the **Cores number** field.
12. As an alternative to entering a username and password, enter the path to the private key file to use for ssh private keyfile authentication or click 🗁 to use the browser to locate the file.

Edit an NLME profile

1. Select **File > Settings**.
2. In the dialog, select the name of the NLME profile from the list on the left.
3. Click 🗑 to delete the selected profile.
4. Modify the settings as needed. (See “Set up profile for NLME” for descriptions of options.)

Software integration

Most of the integration settings are not necessary to define in order to work with Pirana. Many of the program locations are only used to create easy links to the software from within Pirana.

- Extensions integration settings
- Stan integration settings
- PsN integration settings
- R/Xpose setup

Note: Once modifications are made and you have saved them, restart Pirana for the changes to take effect.

Extensions integration settings

Most of the extensions settings are not necessary to define in order to work with Pirana. Many of the program locations are only used to create easy links to the software from within Pirana.

The following are important to set correctly in order to work with Pirana easily:

R location: Pirana uses R and Xpose to generate graphs.

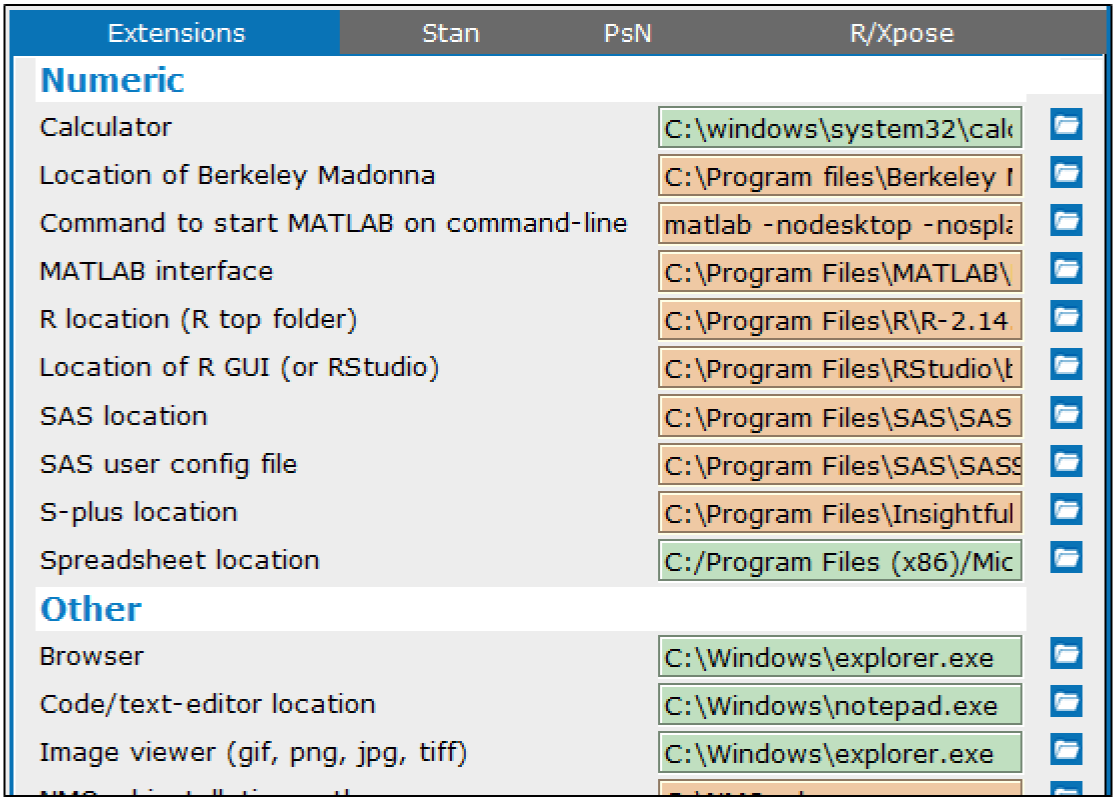
Location of R GUI (if available).

Spreadsheet location (i.e., Excel, Numeric, etc.): Enables viewing of CSV files.

Code/text editor: This editor is used to edit models or scripts.

PDF file viewer: Enables viewing of the many graphics that are created as PDF files.

1. Select **File > Settings**.
2. Select **Software Integration** in the left list.



Numeric

Calculator	C:\windows\system32\calc
Location of Berkeley Madonna	C:\Program files\Berkeley I
Command to start MATLAB on command-line	matlab -nodesktop -nosplá
MATLAB interface	C:\Program Files\MATLAB\
R location (R top folder)	C:\Program Files\R\R-2.14
Location of R GUI (or RStudio)	C:\Program Files\RStudio\t
SAS location	C:\Program Files\SAS\SAS
SAS user config file	C:\Program Files\SAS\SASS
S-plus location	C:\Program Files\Insightful
Spreadsheet location	C:/Program Files (x86)/Mic

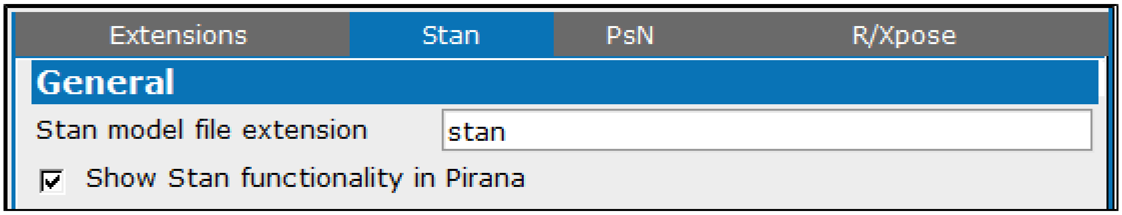
Other

Browser	C:\Windows\explorer.exe
Code/text-editor location	C:\Windows\notepad.exe
Image viewer (gif, png, jpg, tiff)	C:\Windows\explorer.exe

3. In the **Extensions** tab, click the 📂 to browse for and select the location of the listed program **Executables** and **Other** files.

Stan integration settings

1. Select **File > Settings**.
2. Select **Software Integration** in the left list.
3. Go to the **Stan** tab.



General

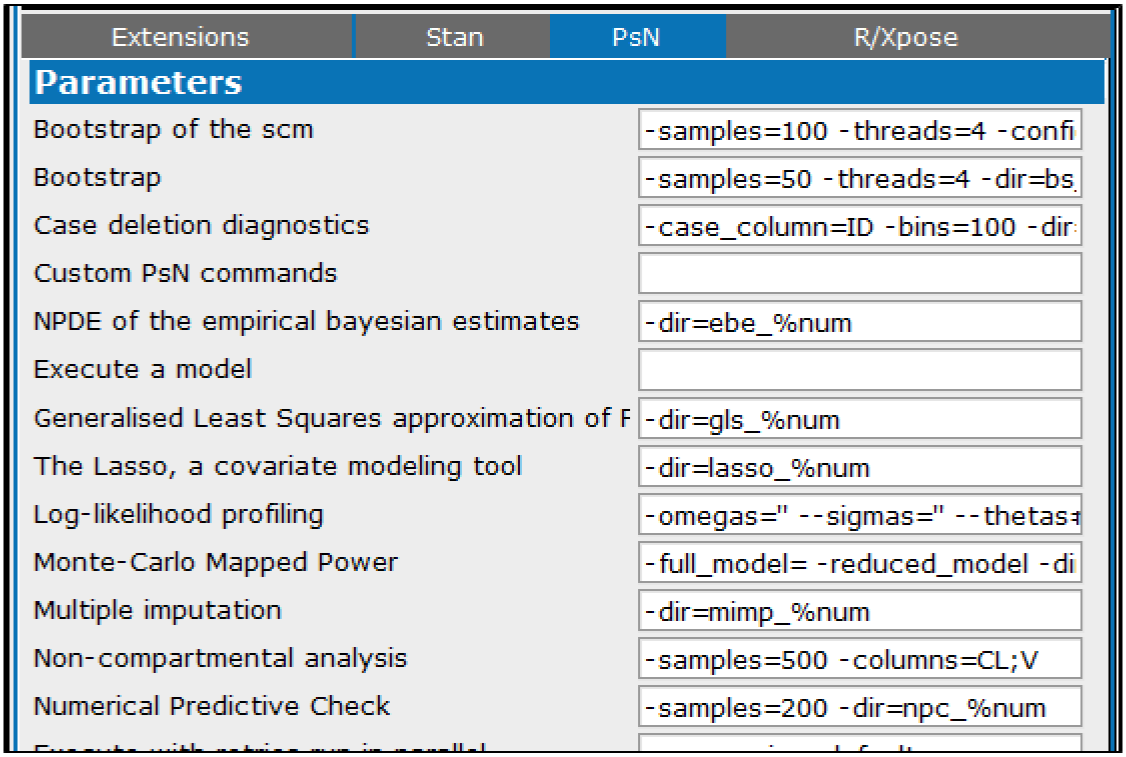
Stan model file extension: stan

☑ Show Stan functionality in Pirana

4. Enter the file extension for Stan model files in the field.
5. Toggle the **Show Stan functionality in Pirana** box to switch between showing and hiding Stan functionality in Pirana’s UI.

PsN integration settings

1. Select **File > Settings**.
2. Select **Software Integration** in the left list.
3. Go to the **PsN** tab.



Parameters

Bootstrap of the scm	-samples=100 -threads=4 -confi
Bootstrap	-samples=50 -threads=4 -dir=bs_
Case deletion diagnostics	-case_column=ID -bins=100 -dir
Custom PsN commands	
NPDE of the empirical bayesian estimates	-dir=ebe_%num
Execute a model	
Generalised Least Squares approximation of F	-dir=gls_%num
The Lasso, a covariate modeling tool	-dir=lasso_%num
Log-likelihood profiling	-omegas='' --sigmas='' --thetas=
Monte-Carlo Mapped Power	-full_model= -reduced_model -di
Multiple imputation	-dir=mimp_%num
Non-compartmental analysis	-samples=500 -columns=CL;V
Numerical Predictive Check	-samples=200 -dir=npc_%num
Execute with retries run in parallel	[illegible]

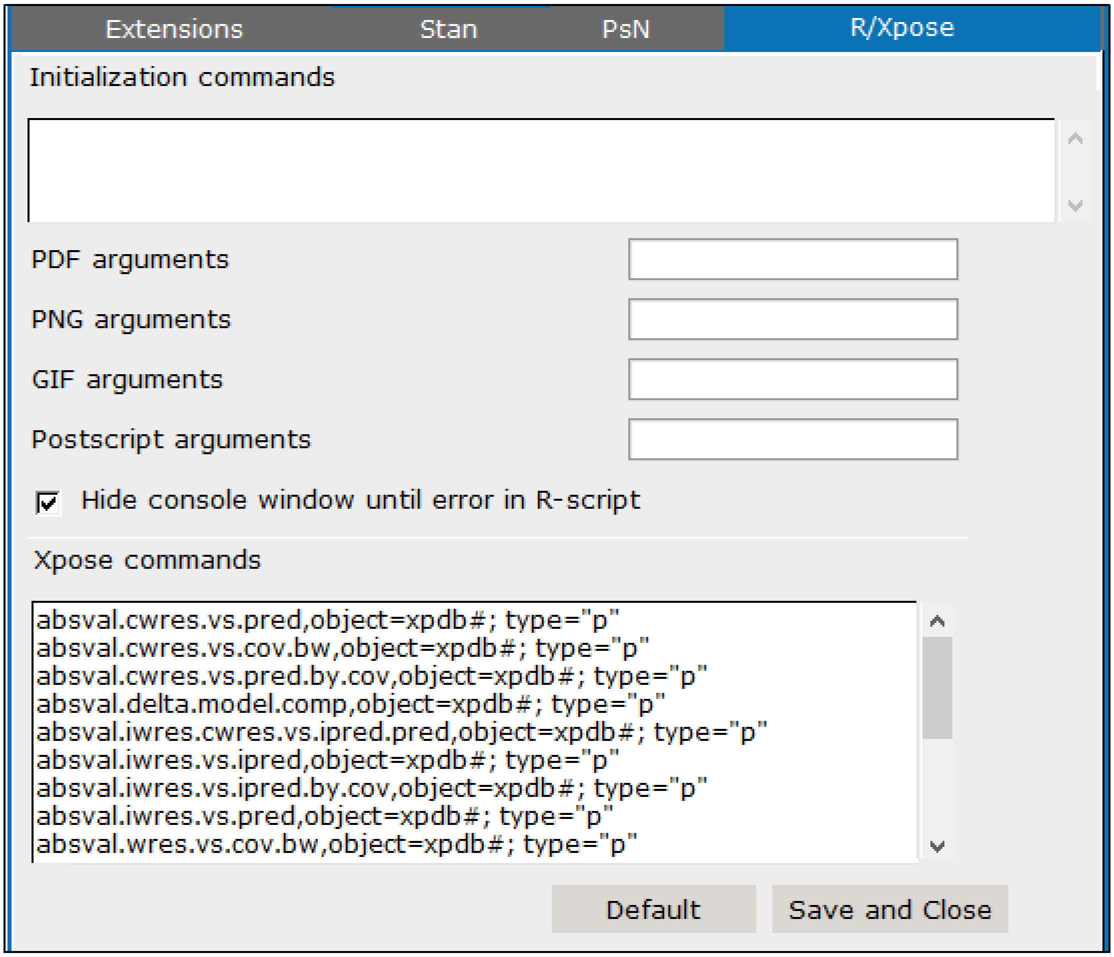
4. Enter the parameter settings to use by default when executing PsN.

   For example, with the Bootstrap parameters set as shown in the previous image, if you selected a run (e.g., run1) and chose **> Model diagnostics > bootstrap**, the default in the **PsN command line** field would be: `bootstrap -samples=50 -threads=4 -dir=bs_run1 run1.mod`

Refer to the PsN documentation for details on specific parameters.

R/Xpose setup

1. Select **File > Settings**.
2. Select **Software Integration** in the left list.
3. Go to the **R/Xpose** tab.



Initialization commands

PDF arguments

PNG arguments

GIF arguments

Postscript arguments

☑ Hide console window until error in R-script

Xpose commands

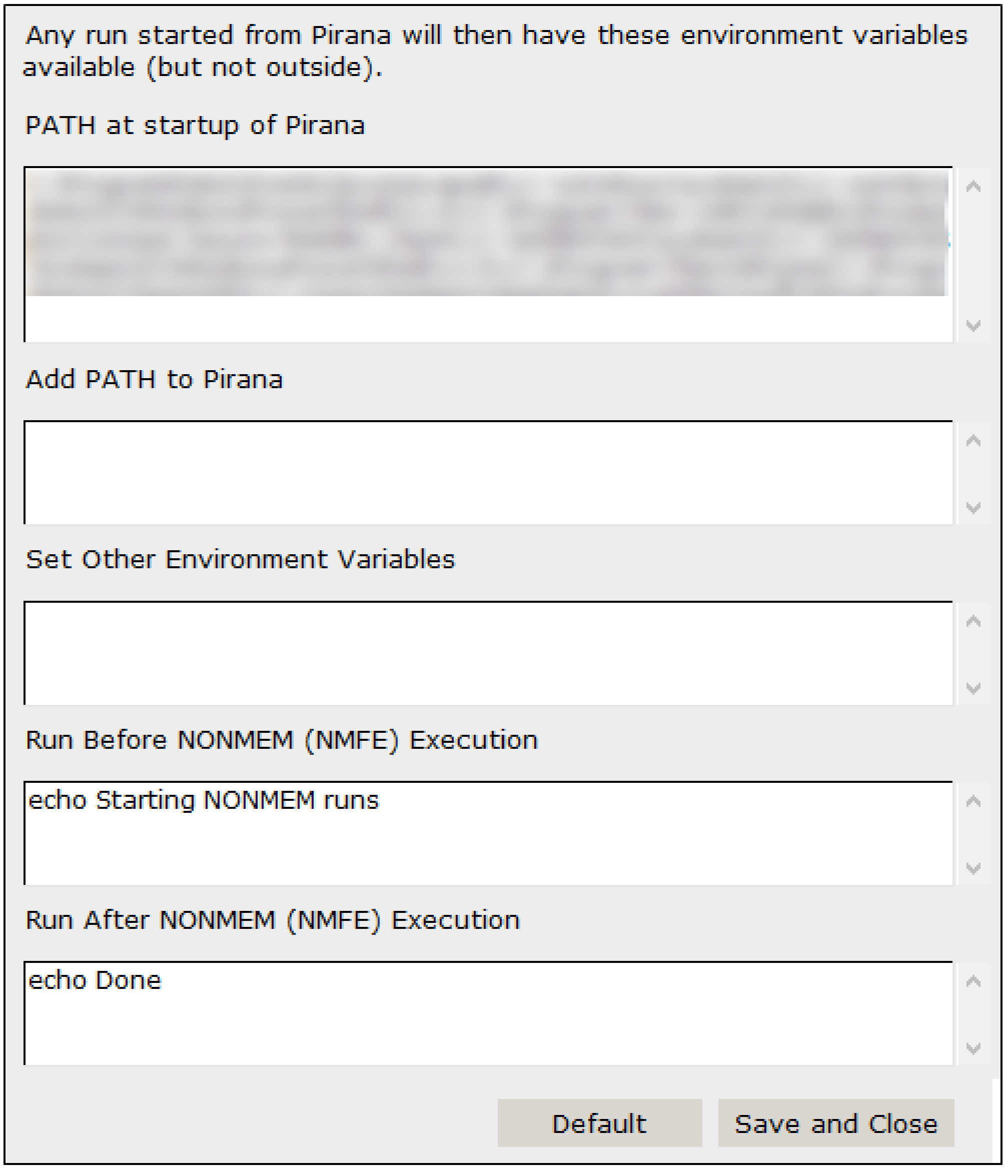
```
absval.cwres.vs.pred,object=xpdb#; type="p"
absval.cwres.vs.cov.bw,object=xpdb#; type="p"
absval.cwres.vs.pred.by.cov,object=xpdb#; type="p"
absval.delta.model.comp,object=xpdb#; type="p"
absval.iwres.cwres.vs.ipred.pred,object=xpdb#; type="p"
absval.iwres.vs.ipred,object=xpdb#; type="p"
absval.iwres.vs.ipred.by.cov,object=xpdb#; type="p"
absval.iwres.vs.pred,object=xpdb#; type="p"
absval.wres.vs.cov.bw,object=xpdb#; type="p"
```

4. Enter any initialization commands for loading specific R libraries when executing R/Xpose in the **Initialization commands** field.
5. In the **PDF arguments**, **PNG arguments**, and **GIF arguments** fields, enter default plotting arguments for R printing devices.
6. Enter any commands to execute at the end in the **Postscript arguments** field.
7. Check the **Hide console window until error in R-script** box to only display the console window if there is a script error.

Note: Changes made in this panel only affect options in the *Run Xpose Commands* dialog (**Results > Xpose > Xpose GUI**). The scripts included with Pirana are not affected.

Environment variables

1. Select **File > Settings**.
2. Select **Environment Variables** in the left list.



Any run started from Pirana will then have these environment variables available (but not outside).

PATH at startup of Pirana

[redacted]

Add PATH to Pirana

Set Other Environment Variables

Run Before NONMEM (NMFE) Execution

```
echo Starting NONMEM runs
```

Run After NONMEM (NMFE) Execution

```
echo Done
```

PATH at startup of Pirana: Shows the PATH Pirana acquired from its environment, e.g., OS or script.

Add PATH by Pirana: Add any locations that affect scripts/applications that are run by Pirana. Instead of modifying the PATH statement, you can just type the locations in this field.

Set Other Environment Variables: Enter any additional environment variables that you want to define.

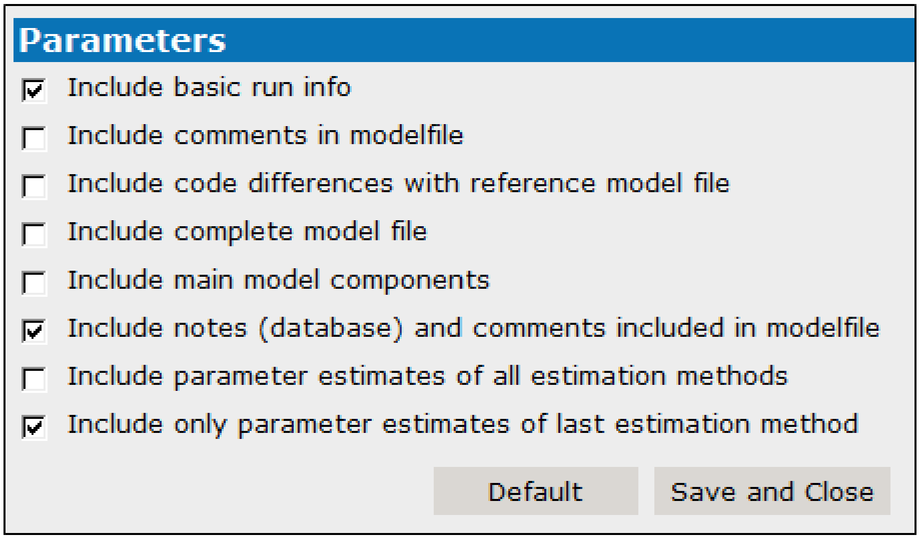
Run Before NONMEM (nmfe) Execution: Enter any commands to run prior to executing NONMEM.

Run After NONMEM (nmfe) Execution: Enter any commands to run after executing NONMEM.

Note: Once modifications are made and you have saved them, restart Pirana for the changes to take effect.

Run reports settings

1. Select **File > Settings**.
2. Select **Run Reports** in the left list.



Parameters

☑ Include basic run info

□ Include comments in modelfile

□ Include code differences with reference model file

□ Include complete model file

□ Include main model components

☑ Include notes (database) and comments included in modelfile

□ Include parameter estimates of all estimation methods

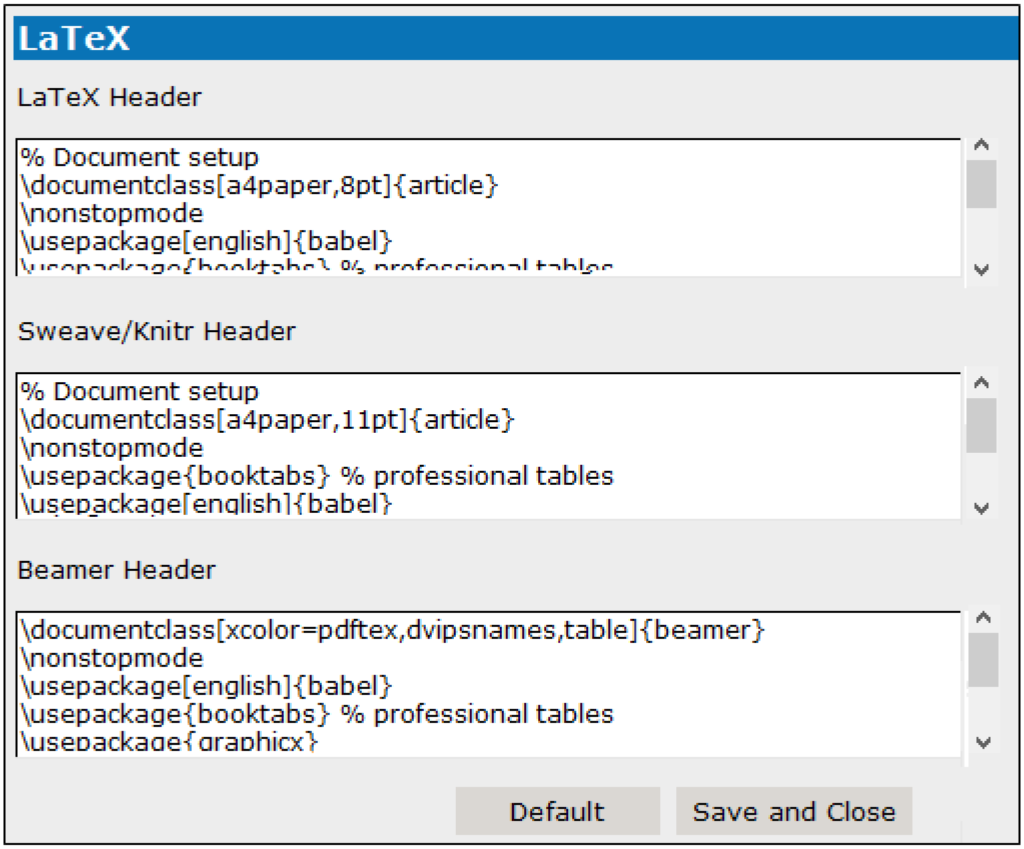
☑ Include only parameter estimates of last estimation method

3. Check the boxes for the items to include in a generated report by default.

Note: Once modifications are made and you have saved them, restart Pirana for the changes to take effect.

LaTeX settings

1. Select **File > Settings**.
2. Select **LaTeX** in the left list.



```
% Document setup
\documentclass[a4paper,8pt]{article}
\nonstopmode
\usepackage[english]{babel}
\usepackage{booktabs} % professional tables
```

```
% Document setup
\documentclass[a4paper,11pt]{article}
\nonstopmode
\usepackage{booktabs} % professional tables
\usepackage[english]{babel}
```

```
\documentclass[xcolor=pdftex,dvipsnames,table]{beamer}
\nonstopmode
\usepackage[english]{babel}
\usepackage{booktabs} % professional tables
\usepackage{graphicx}
```

3. Modify code for the different headers.
   - LaTeX Header
   - Sweave/knitr Header: If the Sweave functionality from Xpose is used, enter the default LaTeX preamble in this field. The preamble specifies what “packages” are to be loaded prior to running the code, along with details about page layout, etc.
   - Beamer Header

Note: Once modifications are made and you have saved them, restart Pirana for the changes to take effect.

Appendices

Keyboard shortcuts

The following keyboard shortcuts are available in Pirana:

Ctrl-R: Run model

Ctrl-L: Open NM output file (`.lst`)

Ctrl-N: New model file

Ctrl-D: Duplicate model file

Ctrl-P: Show parameter estimates for run(s)

Ctrl-T: HTML-file from NM output

Ctrl-E: Execute model (PsN)

Ctrl-B: Bootstrap model (PsN)

Ctrl-V: VPC from folder (PsN)

Ctrl-U: Update inits (PsN)

Ctrl-X: Run Xpose commands

Ctrl-A: Select all models

Ctrl-+/-: Increase/decrease font size

Ctrl-, : Open settings window

F5: Refresh current folder

Command line parameters

- `-console`: Leave console window open. This may be useful when Pirana hangs or crashes, as sometimes an error may be shown on the command line.
- --`nowebserver`: Do not start Pirana webserver.
- --`pirana_lic <lic-file>`: Provide an alternative Pirana license file.
- `-portable`: Use Pirana in portable mode, e.g., from a USB-stick, leave no footprint on computer.
- --`refresh`: Clear Pirana settings on start. May be useful if the settings become broken/inconsistent and prevent Pirana from working properly. User will need to setup everything from scratch.
- --`safe_folder`: Start Pirana UI from home directory.

Troubleshooting/FAQ

Below are some answers to commonly asked questions.

- ***Why the name Pirana?***

  An acronym for: Pirana is a Resourceful Assistant in NONMEM Analysis.

- ***In Pirana's model overview table, I see some models but no results***.

  Check if you have the extensions set correctly in preferences (e.g. `.mod`/`.lst` for model/results files).

- ***Where does Pirana store the notes I make in the model overview?***

  Pirana stores your notes in a database-file (`pirana.dir`) in the current folder. So, if you would install a new version of Pirana, or move your model folder to another place, your notes will not be lost. The database file contains also some other information about the run results (OFV, run success, etc.) and can of course be read out manually as well, using sqlite3.

- ***I would like to cite Pirana in a report or article.***

  Cite as: Pirana x.x Certara USA, Inc., Overlook Center, Suite 101, Princeton, NJ 08540 USA, where x.x is the version number of Pirana.

- **Pirana will not launch after a setting was changed. What can I do?**

  Try one of the following. Be aware that either of these options will ***reset everything*** to original default settings, so ***all custom settings will be lost***.

  – From the commandline, use the -`refresh` option. This clears Pirana settings upon startup. .

  – Delete the *.db file in the `.pirana` directory.