Maximum Likelihood Models User's Guide

Applies to:
Phoenix NLME 8.3

Legal Notice

Trademarks

Certara, L.P.
100 Overlook Center, Suite 101, Princeton, NJ, 08540 USA
Telephone: +1.609.716.7900
www.certara.com

Contents

Maximum Likelihood Models

The Maximum Likelihood Models object allows users to perform a variety of pharmacokinetic and pharmacodynamic analyses using individual and population modeling. It provides access to robust and efficient Maximum Likelihood engines to perform individual, population, and pooled data analyses. Phoenix NLME provides automated covariate selection, bootstrap, and visual predictive check options for population models. It creates consistent graphical and worksheet output to allow easy comparison between models via the Maximum Likelihood Model Comparer object.

Note: Phoenix NLME is only supported on 64-bit systems.

Phoenix NLME is extremely flexible on requirements for input data. There are no requirements for naming a variable as long as it is acceptable in a Phoenix spreadsheet (which does not allow special characters) and is not a reserved word for Phoenix NLME (see “Reserved and user-defined variable names”). Column headers can contain any combination of alphanumeric characters and underscores.

Avoid special characters in the input data. Special characters, such as the Greek letter “beta,” in the data can cause NLME to abort execution.Users have three ways for creating a custom PK model.

The built-in model interface uses the menus in the Maximum Likelihood Models object to create a model, which is customizable to the extent that various options and selections may be combined at the user's discretion. See “Object and built-in model interface” for more information.

The Graphical model interface uses the graphical model editor to create the model structure in diagram form.

The Textual model interface allows users to write their own model using PML (Phoenix Modeling Language), which allows for the greatest amount of flexibility in model structure and customization.

Combinations of these three methods may be necessary to build the desired model. For example, when preparing a recycling model (enterohepatic recirculation), a graphical model can be built, but textual changes must be made to complete the model, such as:

- Add a function, called `switch`, to turn on and off the recycling process.

```
double(Switch)
```

- Adjust the structural model equations to use the Rate variable.

```
deriv(Abile=(A1*K1g)-(Abile*Rate))
deriv(Agut=(Abile*Rate)-(Agut*Ka))
Rate=Switch/Tau #Tau is the gall bladder emptying interval
```

- Add a sequence block to define the gall bladder emptying process.

```
Ri=10            #Time recirculation occurs, 10hrs used here
sequence{
    Switch=0;    #Turn off the gall bladder emptying
    sleep(Ri);   #Wait for 10 hours
    Switch=1;    #Turn on the gall bladder emptying on
    sleep(Tau);  #Wait for the gall bladder emptying interval
    Switch=0;    #Turn off the gall bladder emptying
}
```

- Add fixed effect for Tau.

```
fixef(Tau=c(, 3,))
```

- Comment out any unused parameters and adjust fixed effect values as needed.

Caution: In the NLME interface, where numbers can be entered in data fields, generally either comma (,) or period (.) can be used as a decimal point. (It will be converted to a period.) However, there are fields where sequences of numbers, separated by commas, can be entered, such as the sequence of times in a table specification. In those fields, the comma character cannot be used as a decimal point, because it acts as a delimiter between numbers.

The following symbol in the Phoenix NLME documentation indicates information specific to individual modeling,

This section contains information on the following topics:

Object and built-in model interface

Use one of the following to add the Maximum Likelihood Models object to a Workflow:

Right-click menu for a Workflow object: **New > Modeling > Maximum Likelihood Models**.
Main menu: **Insert > Modeling > Maximum Likelihood Models**.
Right-click menu for a worksheet: **Send To > Modeling > Maximum Likelihood Models**.

Note: To view the object in its own window, select it in the Object Browser and double-click it or press **ENTER**. All instructions for setting up and execution are the same whether the object is viewed in its own window or in Phoenix view.

This section contains information on the following topics:

See "Model output" for lists of result worksheets, plots, and text files.

Main Mappings panel

Use the Main Mappings panel to identify how input variables are used in a Maximum Likelihood Models object. Required input is highlighted orange in the interface.

Note: Context associations change depending on the selected Maximum Likelihood model and on options selected in the Structure tab and Input Options tab.

Note: It is ***not*** a requirement to map dosing information in the Main Setup tab. Dose can be mapped in the Main tab if dosing information columns exist in the dataset that contains the dependent variable. Otherwise, Phoenix NLME has other options for inputting dosing information as described in the Dosing Setup tab section in "Dosing panel".

Columns non-specific to the model type

None: Data types mapped to this context are not included in any analysis or output.

Sort: Up to 5 additional study variables used to sort the output. A separate analysis is performed for each unique combination of sort variable values. For multiple sort variables, set the order for displayed results in the tab 'Output Sort Order'. Do not use sort values when pooling data (see "A note about Pooled data" for more information).

Note: If the number of variables mapped to Sort and ID exceeds five, only the first five will be used. In cases such as this, try using fewer Sort/ID variables or merge them accordingly when possible.

Note: If a variable is used as a sort, it cannot be re-used as a covariate within a model.

Column specific to the population models

ID: Up to 5 categorical variable(s) identifying individual data profiles, such as subject ID and treatment in a crossover study. Only available in Population modeling.

PK model

A1: The amount administered to Compartment 1 for Intravenous (Micro, Clearance, or Macro; **A** for Macro1). Used if **Intravenous** is selected in the **Absorption** menu (Structure tab).

Aa: The amount administered to the absorption compartment for Extravascular (Micro, Clearance, Macro or Macro1). Used if **Extravascular** is selected in the **Absorption** menu (Structure tab).

Time: The relative or nominal dosing times used in a study. See "Time and date variable formatting" for more details.

CObs: The continuous observations of drug concentration in the blood (i.e., the dependent variable). Character and blank (i.e., missing) values are ignored by the model and thus, only numeric entries are taken into account. Negative and non-negative values are accepted. Using a graphical model or a textual model allows the user to have more than two dependent variables in models.

Emax or Linear model

C: The independent variable that is treated as a covariate during the estimation/simulation process. Although this variable is not required by the system, it is likely needed for modeling since it is assumed to be zero if not mapped. Mapped columns can contain character and numeric values (negative and non-negative) and are displayed in the results as IVAR ("independent variable"). Character values are treated as blanks (i.e., missing information) and therefore, values are backwards or forward extrapolated if there is no prior information.

Note: Due to the special embedded nature of the C variable, the posthoc table is not created in the Graphical mode. If required, it is suggested that an explicit covariate C be added in the covariates list.

EObs: The observed drug effect (i.e., the dependent variable). Character and blank (i.e., missing) values are ignored by the model and thus, only numeric entries are taken into account. Negative and non-negative values are accepted. Using a graphical model or a textual model allows the user to have more than two dependent variables in models.

Linked PK, Emax, Indirect, and Linear models all use some combination of the contexts listed above.

Extra input options

A1 Rate/A1 Duration: The rate/duration of drug delivery when using an Intravenous or Infusion delivery method (Micro, Clearance, or Macro; **A Rate** for Macro1). The column **A1 Rate** appears in the Mappings panel when the **Infusions possible?** option is checked in the Structure tab. The column **A1 Duration** appears when the **Infusions possible?** option is checked and then the **Duration?** option is checked.

Aa Rate/Aa Duration: The rate/duration of drug delivery when using an extravascular delivery method. The column **Aa Rate** appears in the Mappings panel when the **Infusions possible?**

option is checked in the Structure tab. The column **Aa Duration** appears when the **Infusions possible?** option is checked and then the **Duration?** option is checked.

Date: Appears in the Mappings panel when **Date?** is checked in the Input Options tab. Year, month, and day are in pre-specified format. See "Time and date variable formatting" for more details.

CObsBQL: Appears in the Mappings panel if **BQL?** is checked in the Structure tab. It allows users to map a dataset to the Main Mappings panel that contains BQL flag for the continuous observation values. See the **BQL?** option description in the Structure tab section for more details.

A0Obs: Appears in the Mappings panel if **Elim. Cpt.?** is checked in the Structure tab. This is the observed amount of drug in the elimination compartment.

MDV: Appears in the Mappings panel when **MDV?** is checked in the Input Options tab. It is used to indicate that there is a missing dependent variable by using a nonzero numeric flag. A zero or blank means that the dependent variable value is present. This flag is optional for Phoenix NLME as the tool recognizes which records contain observed values and which do not. However, it can be optionally used, for example, to exclude observations from certain analyses.

A summary of the rules that determine which records are used in Phoenix NLME are as follows:

- If a single data sheet (combined dosing and observations), mapped from the Main panel, has a dose and a nonzero observation appearing on the same line, then the nonzero observation is treated as real data and is not ignored.
  (Except if there is an explicit nonzero MDV entry on the line, in which case the observation is always ignored).
- If a dose and an observation of 0 appear together on the same line, then the observation is ignored.
  (Except for the categorical observation case, where the observation is not ignored, unless there is an explicit MDV entry of 1 on the line).
- If the dosing information is mapped from the Main panel, there is no associated dose record, and no MDV entry or no MDV entry greater than zero, then an observation with a value (including a value of zero) is a valid observation.
- If the dosing information is entered using the dosing panel (either manually in an internal worksheet or from a file containing dosing information), then the dosing information is not merged in the same line as the observation information, even if the time entries are the same. In this case, all observations will be used unless there is an explicit nonzero MDV flag on the observation line.

SteadyState: Appears in the Mappings panel when **Steady State** is checked in the Input Options tab. It is used if the model reaches a steady state of dosing. See "More on Steady State" for more details.

ADDL: Appears in the Mappings panel when **ADDL** is checked in the Input Options tab. It is used if additional identical doses are included in the dataset. See "More on ADDL" for more details.

A note about Pooled data

The NLME engine used by the Maximum Likelihood Models object can be used for analyzing pooled data. This approach fits one set of model parameters to all individuals in a dataset. There are some requirements that must be met to properly configure a pooled model:

- Uncheck the **Population** checkbox, if it is checked.
- Do not map any profile data for individuals. Using profile data causes multiple models to be estimated.

- If the model includes differential equations, such as the "deriv" statements in the Phoenix Modeling Language:
  - Make sure that the data are sorted by individual then by time, so that all the observations for any individual are in consecutive and ascending time order. Then clear the **Sort Input?** checkbox on the Run Options tab.
  - Select the **Reset?** checkbox in the Input Options tab and include a column with an indicator for the initial time.

  Dosing information must be placed in the observation data. The Dosing Panel does not function properly for Pooled data.

Time and date variable formatting

A time variable (i.e., independent variable) is required for time-based models (e.g. PK models, or any model capable of undergoing time-evolution). Columns mapped as Time must contain numeric values or character entries in an accepted time format.

The values in the Time column are evaluated to decide if time is complex or not. Time is considered complex if there is a Date column, or if the Time column contains anything indicating clock time, like a colon (:), "am", "pm", "a", "p", etc. If time is complex, it means time within the subject is measured relative to the first time in that subject, it is not absolute. If time is not complex (i.e., simple) it means time is absolute, so even if the first event in a subject occurs at time 10, the subject starts evolving at time 0, and evolves for 10 time units before getting to that first event. If a dataset contains blank time data, the model will not execute.

Phoenix NLME uses the basic time format of hour:minute:second. Below are some examples of acceptable variations of this format:

hh:mm:ss (24 hour clock, e.g., 15:00:00)

hh:mm:ss tt (where tt can be 'am', 'pm', 'AM' or 'PM' in any mixture of upper and lower case, e.g., 03:00:00 pm)

hh:mm:ss t (where t can be 'a','p', 'A' or 'P', e.g., 03:00:00 p)

hh:mm (seconds are assumed 00, e.g., 15:00 or 03:00 pm or 03:00 p)

This format (time with a ':' character) is converted into hours (basic unit of time) although no unit is printed in the output. If there are units in the time column these are not taken into account for this format.

Formats like hh:mm also accept non leading zeros so both 03:00 and 3:00 are accepted. Similarly, both 03:01 and 3:1 are accepted.

Hours are not limited to 24, and minutes and seconds are not limited to 60.

Hour or hour.fraction

Numeric times are accepted (integers and fractions). These can be used by themselves as relative times from first event, or in combination with a date variable. If they are combined with a date variable, then the dosing date/time will be subtracted from the sample date/time to calculate a relative time. For example, if the dose is given on 01/01/2010 at time 2 and the first and second samples are taken at 01/01/2010 at time 12.5 and 01/02/2010 at time 2 respectively, then the relative times used would be a dose at time zero and samples at times 10.5 and 24.

If there are units on a time column that contains numeric values without a date variable, these units are carried to the model output.

A Date format can be specified by checking the option of **Date?** and selecting an appropriate format in the Input Options tab. This option indicates that event times are a combination of a date and a time

format. Note that when selecting a date format, the user needs to map two columns: one for date and one for time.

The accepted date formats in Phoenix NLME are:

Day-Month-Year (most of world)

Month-Day-Year (U.S.)

Year-Month-Day (Asia, ISO)

Year-Day-Month

The date consists of one, two, or three numbers separated by any non-numeric character (4 26 10, 2010/26/04, etc.). If there are three numbers, they are assumed to be the year, month, and day in whatever order the user has chosen. If there are only two numbers, they are assumed to be month and day. If only one number, it is assumed to be the day.

If the year is four digits long, it is taken as is. If the year is three digits long, it is assumed to be in the millennium starting at 1980. (000=2000, 979=2979, 980=1980). If the year is two or one digits long, it is assumed to be in the century starting at 1980. (99=1999, 00 or 0=2000, 79=2079, 80=1980). Leap years are assumed to occur in 1980 and every fourth year after. Leap seconds are not considered. If the year is not given, it is assumed to be 1980. This is the number called the “CenturyBase” and it is 1980, and cannot be changed in the User Interface although it can be changed in command-line mode.

Months are numeric, starting with 1 for January. If a month number is less than one or greater than 12, it is flagged as an error

Days of the month start with 1. If a day number is less than 1 or greater than 31, it is flagged as an error. The number of days in a month depends on the month, and for February, it depends on the leap year.

Dates are converted into the number of hours since 00 hours January 1, 1980.

A non-existent date, such as 1981/02/29 or 09/31 is not flagged, but just wraps into the following month.

Any combination of these time/date formats is accepted by Phoenix NLME.

Although Phoenix worksheets accept other date formats as input (see “Date and time formats” in the Phoenix Worksheets section), only the four date formats above are accepted as time entries in Phoenix NLME.

Phoenix NLME does NOT require that time entries be sorted. By default, Phoenix NLME will sort the time entries within ID. This default can be overwritten by unselecting the **Sort Input?** selection under the Run Options tab (e.g., when executing pooled data models). If **Sort Input?** is unchecked, and time entries are not sorted in the input data, the model will not run.

Columns mapped as Time must be numeric values or text entries in an accepted time format. If the **Date?** option is selected, the user would then need to map both Date and Time columns in one of the accepted formats described above.

More on Steady State

The steady-state flag indicates whether the associated dose is a steady-state dose. When an observed dose is flagged as a steady-state dose, this dose is assumed preceded by a series of dose cycles given at certain regular time intervals such that they have reached steady state (i.e., PK equilibrium). The doses preceding the observed dose have not been captured in the dataset and thus the information has to be entered. The steady-state flag and the 'implied' dosing information indicate to the program that the formulas to be used need to compute the steady-state amounts at the time the steady-state dose is introduced.

The steady-state flag needs to be numeric:

- Values of zero or blank indicate that the record is simply an ordinary dose.
- Values greater than zero indicate that prior to giving the indicated dose, the PK model is put into steady state.

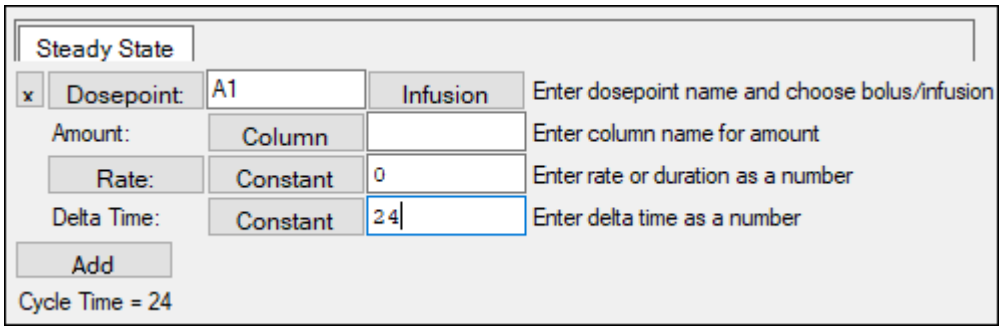
When a record is flagged as being at steady state the program needs some additional information about the dosing cycle when this steady-state observation was taken. Checking the **Steady State** checkbox in the Input Options tab adds a SteadyState column in the Main and Dosing mappings panels and a Steady State sub-menu in the tab for entering the additional dosing information is displayed.

Dosepoint/Dosepoint2: The mode of administration (either bolus or infusion) and the name of the compartment in which the dose is administered (e.g., default name 'Aa' for a one-compartment oral administration). The user must enter this dosepoint name, therefore, the name needs to exist in the model. All available dosepoint names will be listed in the Main mappings panel.

Amount: The dose amount given at each time interval during the cycle. If the amount can be entered manually and is assumed constant during a dosing cycle, then the **Constant** button should be selected and the dose amount entered. If the dosing cannot be assumed constant or the user wishes to enter it from an input file, then click on the toggle button, select **Column**, and enter the name of the column that contains this information. Ensure that this column (containing steady-state dose amounts) exists in the input dataset that contains the dose information for the observed record.

Rate/Duration: If the dosepoint route of administration is selected as an Infusion, then a rate or duration of infusion needs to be provided. This value should be a positive number and the units should be in accordance to time units in the input dataset.

If the rate/duration of infusion is assumed constant during a dosing cycle then the **Constant** button should be selected and rate/duration entered in the field. If the value cannot be assumed constant or the user wishes to enter it from an input file, then click on the toggle button to select **Column** and enter the name of the column containing this information. Ensure that this column (which contains the rate/duration of infusion) exists in the input dataset that contains the dose information for the observed record.

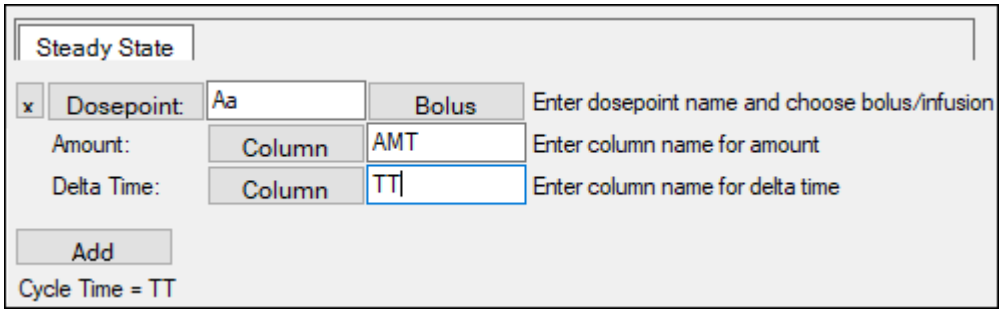


Delta Time: The time interval between doses that allowed steady state to be reached. If the elapsed time between 'implied' doses can be assumed constant during a dosing cycle then the **Constant** button should be selected and the time interval typed in the field. If the delta time cannot be assumed constant or if the user wishes to enter it from a file, then click on the toggle button to select **Column** and enter the name of the column that contains this information. Ensure that this column (containing steady-state delta time) exists in the input dataset that contains the dose information for the observed record.

The dosing cycle can contain multiple steps and doses to multiple compartments. This is specified by adding segments to the cycle, by clicking the **Add** button. The total length of the cycle is the sum of the Delta Time parts of each segment. In steady state, in each segment of the cycle, the dose occurs first, followed by the delta time.

Only numeric values or blanks are accepted in columns mapped to SteadyState. A value of zero or blank indicates that the observation is NOT collected at steady state. Any other value indicates that the model is put into steady state before administering the dose on that row.

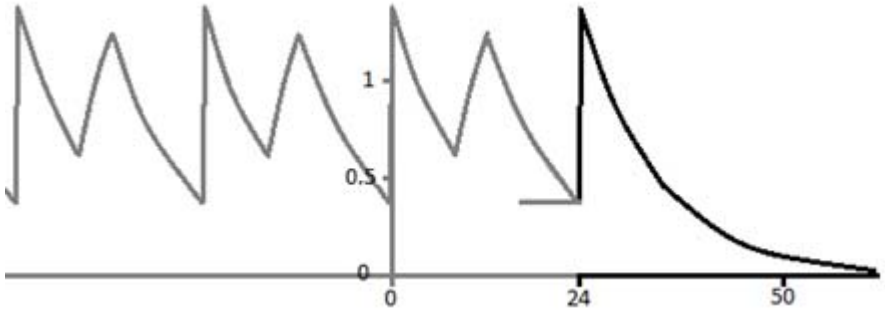
Example: The selections for steady state depicted below indicate that an infinite number of bolus doses are assumed added to the amount in the absorption compartment in the model (Aa) for that subject, in the amounts provided in the AMT column in the dataset, at time intervals provided in the TT column in the dataset. Therefore, for Subject 1, the data are assumed collected at steady state following a dosing regimen of 100 oral units every 24 hours, with the most recent dose administered 24 hours ago.



SUBJ	AGE	BW	CLCR	SEX	RACE	TIME	AMT	CONC	SS	TT
1	32.4623	53.1672	97.642	1	4	288	100	.	1	24
1	32.4623	53.1672	97.642	1	4	312	.	.	.	.
1	32.4623	53.1672	97.642	1	4	312.748	.	1.19456	.	.
1	32.4623	53.1672	97.642	1	4	314.748	.	0.282537	.	.
1	32.4623	53.1672	97.642	1	4	318.053	.	0.10009	.	.
1	32.4623	53.1672	97.642	1	4	480.066	.	0.945309	.	.
1	32.4623	53.1672	97.642	1	4	481.218	.	0.727491	.	.
1	32.4623	53.1672	97.642	1	4	482.886	.	0.194011	.	.

More complex example: Suppose the volume of a 1-compartment model is 1. Suppose it is desired to say that at time 24, one can assume a continuous series of dosing cycles taking place, where the cycle consists of first a bolus dose of 1 unit, and then 8 hours later a 4-hour infusion of 1 unit is given. Then the next cycle starts 24 hours after the initial bolus. Also suppose at time 24 a single bolus of 1mg is given.

The following plot shows concentration versus time. Notice that at time 24, when the actual dose is given, the prior concentration is not zero. It is about 0.4, because of the prior assumed dosing cycles, shown in light gray.

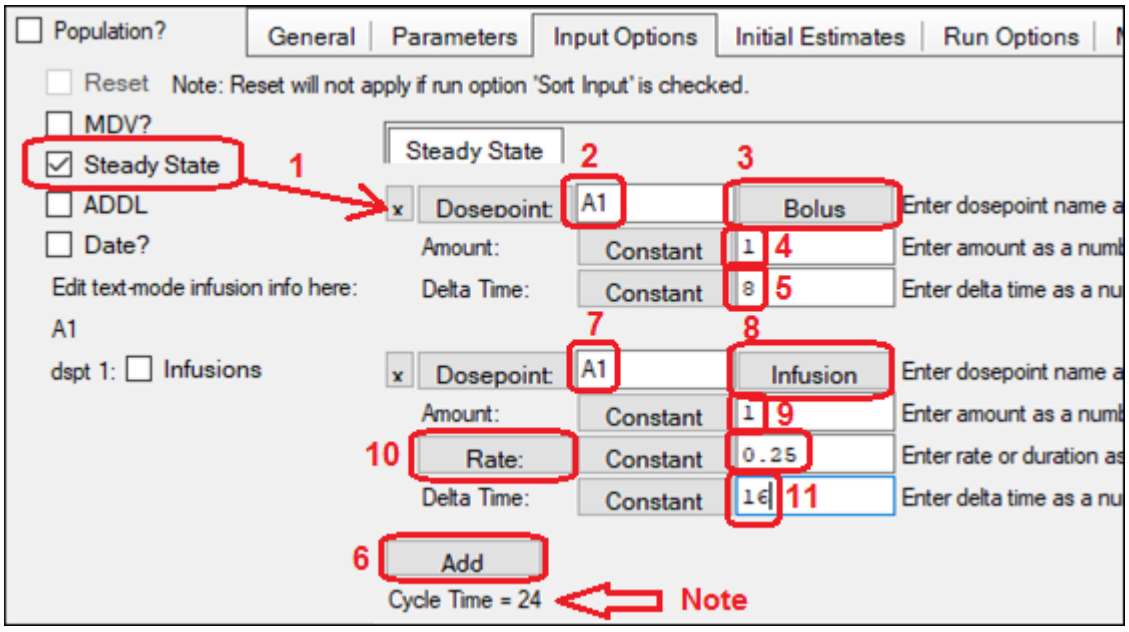


This situation is set up as follows:

1. On the Input Options tab, check the **Steady State** option to display the Steady State sub-tab.

2. Enter the name of the dosepoint to receive the dose, if necessary (e.g., A1).
3. Set up the initial bolus by making sure the Bolus/Infusion option is set to **Bolus**.
4. Enter the amount of the dose (e.g., 1 unit).
5. Enter the wait time before the next dose (e.g., 8 time units).
6. Click the **Add** button to enter the second dose.
7. Enter the name of the dosepoint (which does not have to be the same as the prior dosepoint).
8. Make sure the Bolus/Infusion button says **Infusion**.
9. Enter the amount of the infusion (e.g., 1 unit).
10. Make sure the Rate/Duration button says **Rate** and enter the rate (e.g., 0.25 drug units per time unit).
11. Enter the time that passes before the cycle repeats (e.g., 16 time units).

Note the total cycle time appears below the **Add** button.



The concentration-time curve in the graph shown earlier in this section arose from the following dosing input:

A1	A1 Rate	Time	SteadyState
1	2	24	1

This says two things:

- At time $t=24$ the model is put into steady-state on the assumption that the dosing cycle above occurred at $t-24$, $t-48$, etc., but ***not*** at time t.
- A bolus dose of 1 unit is given at time t.

Note that the delta-time comes after the dose. The reason for this is that the dose may be an infusion; it may have a tlag; it may be zero-order (implied infusion). These delayed aspects of the dose have to be completed before the dosing cycle completes. This might be considered a limitation of the software, but it is necessary to assume that if a subject is in steady-state prior doses have actually completed.

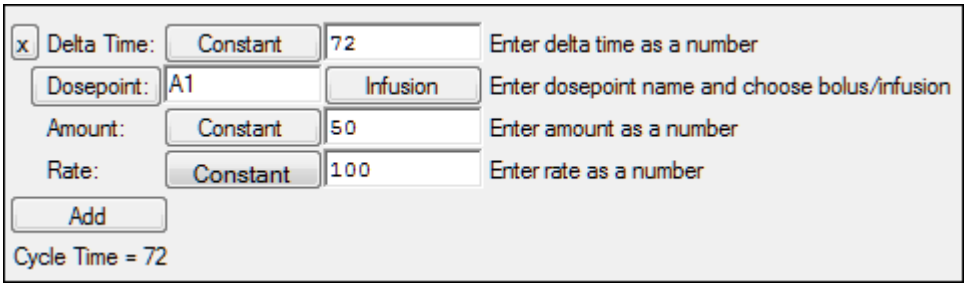
More on ADDL

ADDL can be regarded as a number of additional dose(s) that were not observed but were administered. For example, this can be used to indicate that additional doses will be given after the observed dose. Additional dose information needs to be provided and will be applied each time to the number of additional doses indicated by the ADDL record.

When **ADDL** is checked in the Input Options tab, an ADDL column is added in the Main and Dosing mappings panels and an ADDL sub-menu for entering the additional dosing information is displayed in the tab. The information requested will be the **Dosepoint** (mode of administration and the name of the compartment in which the dose is administered), the **Amount**, the **Rate** if an infusion was administered, and **Delta Time** (i.e., the time interval between doses). The data structures for these options are exactly the same as described in the steady-state section above. However, there is one major difference between ADDL and Steady State, even though the data structure requirements are the same. In Steady State, the dose occurs first, followed by the delta time; in ADDL, the delta time occurs first, followed by the dose.

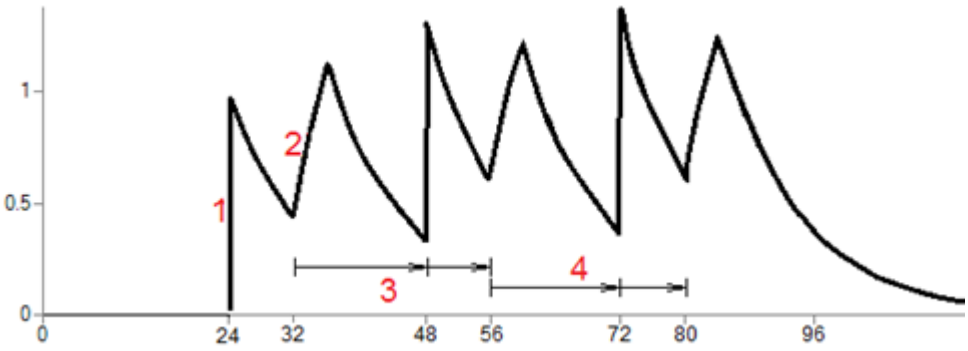
Only numeric values or blanks are accepted in columns mapped to ADDL. A blank value is ignored; a positive numeric value indicates the number of additional doses given, which need additional information, namely, Dosepoint, Amount and Delta Time.

Example: The screen shot below is an ADDL set up to simulate a dosing regimen. For Subject ID 10, the simulated regimen consists of four infusions of 50 mass units at a rate of 100 mass units per time unit (i.e. duration=0.5 time units) every 72 time units (e.g., hours), starting at time zero.



	TID	ID	TIME	DV	RATE	AMT	MDV	GEN	AGE	WT	ADDL
1	1	10	0	0	92	46	1	0	3	15.6	4
2	1	10	1.5	6.37	0	0	0	0	3	15.6	0
3	1	10	5	2	0	0	0	0	3	15.6	0
4	1	10	7	1.47	0	0	0	0	3	15.6	0
5	1	10	8	1.41	0	0	0	0	3	15.6	0
6	1	10	10	1.06	0	0	0	0	3	15.6	0
7	1	10	13	0.604	0	0	0	0	3	15.6	0

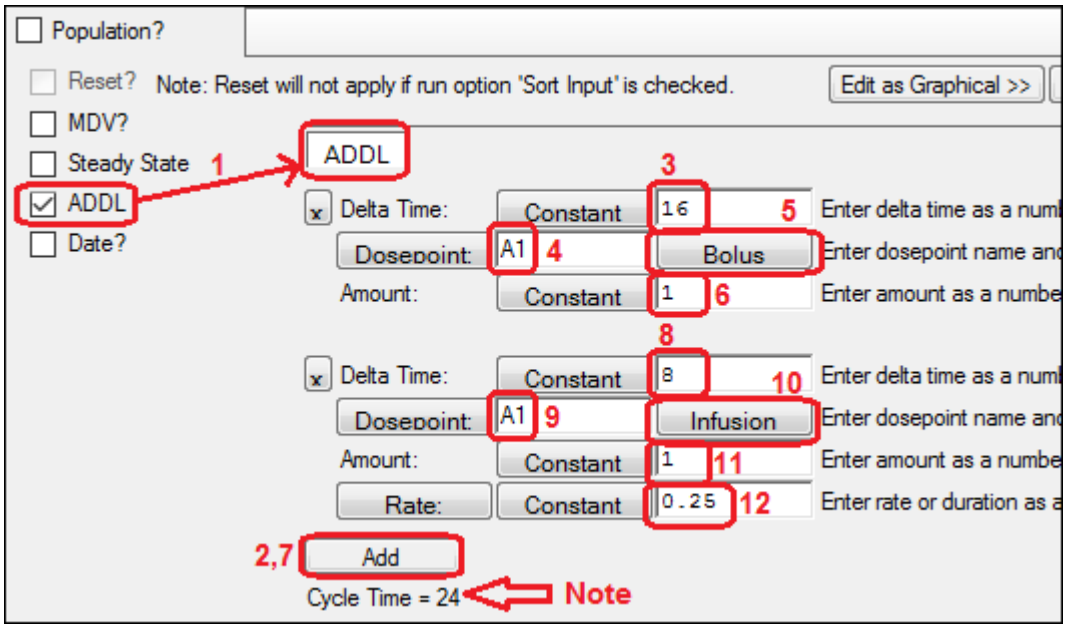
More complex example: Suppose a bolus dose of 1 unit given at time 24 (area 1 in the following graph), followed by a 4-time-unit infusion (area 2) started at time 32 (8 hours after the bolus), followed by two additional cycles (areas 3 and 4) just like it.



The additional cycles have to start at the time of the last actual dose, which is the infusion that starts at time 32. So the cycle, as depicted in area 3 for example, starts with an initial delay of 16 time units, then the bolus, then a delay of 8 time units, then the start of the next infusion. This cycle is repeated as many times as requested, in this case 2 times.

- On the Input Options tab, check the **ADDL** option to display the ADDL sub-tab.
- Click the **Add** button.
  Notice how ADDL differs from the Steady-State user interface. The **Delta Time** delays come first.
- Enter the initial delay (e.g., 16 time units).
- Enter the name of the dosepoint to receive the dose (e.g., A1), if necessary.
- Make sure the Bolus/Infusion option is set to **Bolus**.
- Enter the amount of the dose (e.g., 1 unit).
- Click the **Add** button to enter the second dose.
- Enter the initial delay (e.g., 8 time units).
- Enter the name of the dosepoint (which does not have to be the same as the prior dosepoint).
- Make sure the Bolus/Infusion button says **Infusion**.
- Enter the amount of the infusion (e.g., 1 unit).
- Make sure the Rate/Duration button says **Rate** and enter the rate (e.g., 0.25 drug units per time unit).

Note that the total cycle time is shown below the **Add** button.



The concentration-time curve shown in the previous graph arose from the following dosing input:

A1	A1 Rate	Time	ADDL
1	1	24	
1	2 0.25	32	3 2

This says three things:

- A simple bolus of 1 unit occurs at time 24.
- A 4-time-unit infusion begins at time 32 (8 hours after the bolus).
- Two of the ADDL cycles begin at time 32.

Since the first cycle begins with a 16-time-unit delay, it means that the bolus in the cycle occurs at time 48, which is 24 hours after the initial bolus called for in the data.

Specifying steady state and additional doses outside of the user interface

Steady state dosing can be specified outside of the user interface. Consider the following data from an input file.

	Id	time	SSOffset	A1	Aa	SS	II
1	1	0		100	30	1	8
2	1	0	3	50	60	2	8

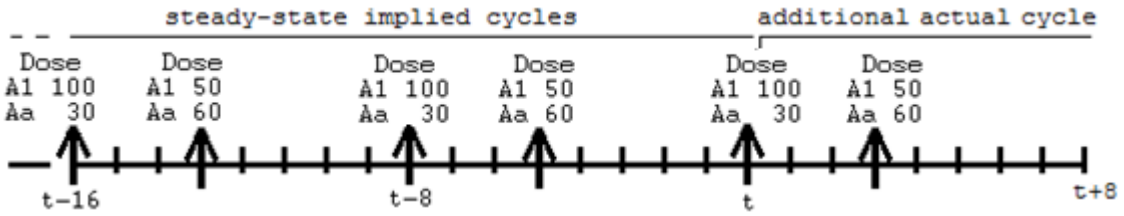
The column definition text file contains these lines:

```
sscol(SS)
iicol(II)
ssoffcol(SSOffset)
```

where `sscol(SS)` indicates that there is a column named SS (meaning steady state), and `iicol(II)` indicates that there is a column named `II` (meaning interdose interval). The SS column contains values 1 or 2 (or nothing). Note, rows in which SS is 2 can only follow rows in which SS is 1 or 2.

When the first row is encountered, at time 0, it means that the model will undergo a cycle of `II` time units (in this case 8), where each iteration of the cycle consists of a dose of 100 to A1, and a dose of 30 to Aa, followed by 8 hours until the next repetition. This cycle is conceptually repeated sufficient times to bring the model to steady state, ending at time 0, because 0 is the time on the row. Then, at that time, the doses are repeated one more time.

Multiple cycles can be superimposed, as shown by the second row, in which SS is 2, indicating an 8 time-unit cycle with a dose of 50 to A1 and 60 to Aa. However, the doses are not given at time 0. They start at time 3, because there is a datum 3 in the `SSOffset` column. Therefore, the cycle looks like this:



Up to nine SS=2 rows can follow an SS=1 row. In addition, the interdose interval (`II`) values do not all have to be the same. However, the least common multiple of the interdose intervals must not be more than ten times the longest one. This prevents combinations of interdose intervals like 24 and 23 (for which the least common multiple would be 552 time units).

Note on transferring NONMEM datasets to NLME, and vice-versa: In the case that a NONMEM dataset contains rows in which SS=2, and in which the time value on that row differs from the SS=1 row above it, it is necessary to introduce an additional column for SSOffset:

	Id	time	A1	SS	II
1	1	10	100	1	12
2	1	13	50	2	8
3	1	16	50	2	8

Note the different values in the time column.

	Id	time	SSOffset	A1	SS	II
1	1	10		100	1	12
2	1	10	3	50	2	8
3	1	10	6	50	2	8

Note the same values in the time column with the additional offset column data (SSOffset).

This was necessitated by the Sort option on input data.

Additional doses can be specified as follows:

	Id	time	A1	Aa	II	ADDL
1	1	0	100	30	8	10
2	1	3	50	60	8	10

The column definition text file contains these lines:

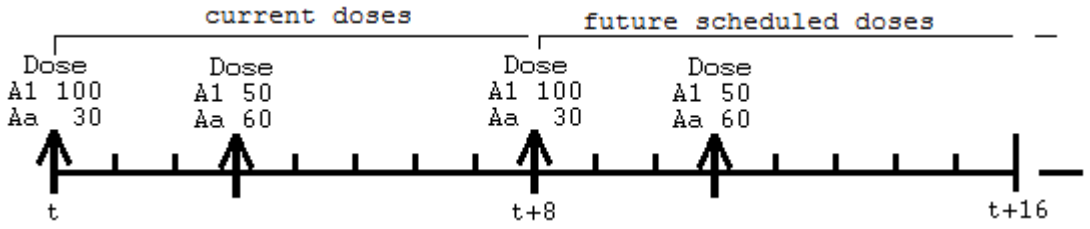
```
addlcol(ADDL)
iicol(II)
```

where `addlcol(ADDL)` indicates that there is a column named ADDL (meaning additional), and `iicol(II)` indicates that there is a column named `II` (meaning interdose interval).

When the first row is encountered, at time 0, first the doses into A1 of 100 and Aa of 30 are performed, and then 10 more of those doses are scheduled into the future, at 8 time-unit intervals, because ADDL=10 and `II`=8.

Then, when the second row is encountered at time 3, first the doses into A1 of 50 and Aa of 60 are performed, and then 10 more of those doses are scheduled into the future, at 8 time-unit intervals.

Together the ADDL doses look like this:

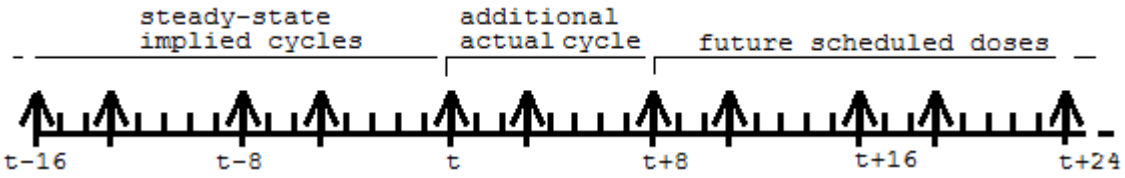


SS and ADDL dosing may be combined, as in the following:

	Id	time	SSOffset	A1	Aa	SS	II	ADDL
1	1	0		100	30	1	8	10
2	1	0	3	50	60	2	8	10

where the column definitions are:

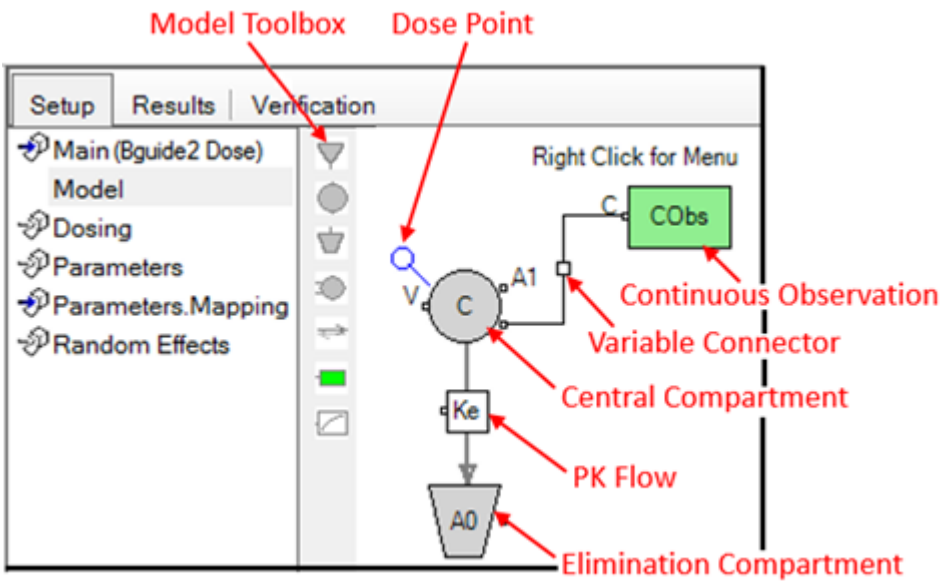
```
sscol(SS)
iicol(II)
ssoffcol(SSOffset)
addlcol(ADDL)
```



Covariates may be included on SS or ADDL lines, but observations will be ignored.

Model panel

The Model panel is utilized with the graphical or textual building of models. It is not used for built-in models. However, it does show a graphical representation of a built-in model, if possible. In some cases, it is not possible to show a unique model diagram; for example, in the case of closed-form macro constant models.



Dosing panel

The Dosing panel is useful for PK or PK/PD models in the case where the dosing information and observations are in different files.

None: Data types mapped to this context are not included in any analysis or output.

Sort: Up to 5 additional study variables used to sort the output. A separate analysis is performed for each unique combination of sort variable values. Do not use sort values when pooling data. See "A note about Pooled data" for more information.

ID: Up to 5 categorical variable(s) identifying individual data profiles, such as subject ID and treatment in a crossover study. Only available in Population modeling.

Note: If the number of variables mapped to Sort and ID exceeds five, only the first five will be used. In cases such as this, it is suggested to use less Sort/ID variables or merge them accordingly when possible.

A1: The amount administered to Compartment 1 for Intravenous (Micro, Clearance, or Macro). Used if **Intravenous** is selected in the **Absorption** menu (Structure tab).

A1 Rate/A1 Duration: The rate of infusion for Intravenous or Infusion (Micro, Clearance, or Macro; **A Rate** for Macro1). The column **A1 Rate** appears in the Mappings panel when the **Infusions possible?** option is checked in the Structure tab. The column **A1 Duration** appears when the **Infusions possible?** option is checked and then the **Duration?** option is checked.

Aa: The amount administered to the absorption compartment for Extravascular (Micro, Clearance, Macro or Macro1). Used if **Extravascular** is selected in the **Absorption** menu (Structure tab).

Aa Rate/Aa Duration: The rate/duration of drug delivery when using an extravascular delivery method. The column **Aa Rate** appears when the **Infusions possible?** option is checked in the Structure tab. The column **Aa Duration** appears when **Infusions possible?** is checked and then **Duration?** is checked.

Time: The time of dose administration. See "Time and date variable formatting" for acceptable formatting.

Selecting different input options in the Input Options tab adds the following extra columns in the Dosing panel for each model.

MDV: Appears when **MDV** is checked in the Input Options tab. Missing dependent variable flag could be mapped. See the MDV description under Main Mappings panel for more information.

Steady State: Appears when **Steady State** is checked in the Input Options tab. Used if the model reaches a steady state. See "More on Steady State" for more details.

ADDL: Appears when **ADDL** is checked in the Input Options tab. Used if additional identical doses are included in the dataset. See "More on ADDL" for more details.

Date: Appears when **Date** is checked in the Input Options tab. Year, month, and day are in a prespecified format. See "Time and date variable formatting" for acceptable formatting.

Note: If an internal worksheet is used, input for all values of the sort variables is required.

Note: Rebuild an internal dosing worksheet when changing a model from individual to population, or vice versa, and changing between Sort and ID mapping. Otherwise, it can cause a verification error upon execution.

Parameters panel

The Parameters panel allows users to map a worksheet with initial, lower, and upper parameter values for fixed effects. This is a convenient way to use results from another computation to set initial estimates for a model without manually entering values.

Parameter value options can also be specified in the Fixed Effects tab, which is located under the Parameters tab. If the Parameters panel is used, the values in the Fixed Effects tab are overridden.

None: Data types mapped to this context are not included in any analysis or output.

Sort: Up to 5 categorical variable(s) identifying individual data profiles, such as subject ID and gender. A separate analysis is done for each unique combination of sort variable values. It should be the same as the variables set in the Main panel.

Parameter: The parameters used in the structural model.

Initial: The initial value for each parameter.

Lower: The lower limit value for each parameter.

Upper: The upper limit value for each parameter.

Use the **Rebuild** button to reset the internal worksheet to its default state.

Note: If an internal worksheet is used, input for all values of the sort variables is required.

Note: If only partial estimate information is entered in the Parameters grid (i.e., some estimates are blank), the program will use the estimates from the previous profile.

Parameters.Mapping panel

If an external worksheet is mapped to the Parameters panel, and the parameter names in the worksheet do not match the parameter names in the model, the Parameters.Mapping panel can be used to match external parameter names with internal ones. If the parameter names match or no external parameters worksheet is used, then there is no need to use the Parameters.Mapping panel.

- Map an external worksheet with parameters and initial estimates to the Parameters panel.
- Select the **Parameters.Mapping** panel.
  The **Use Internal Worksheet** checkbox is selected by default.
- Click **Rebuild** to update the internal Parameters.Mapping worksheet.
- Beside each external parameter name in the *Names* dialog, select the checkbox underneath the corresponding internal parameter name.
- Click **OK** to accept the matched parameter names and close the dialog.

Note: If the Parameters.Mapping panel is not used when a worksheet is specified for initial estimates, the values shown in the Parameters tab are used.

Random Effects panel

The Random Effects panel allows users to map a worksheet with random effects posthoc values.

For non-textual mode, it is activated for PKPD models when the **Sequential PK/PD** checkbox is enabled – this will freeze the PK portion of random effects and turn them into covariates. For textual mode, all covariates presented in the model could be mapped.

None: Data types mapped to this context are not included in any analysis or output.

Sort: Up to 5 categorical variable(s) identifying individual data profiles, such as subject ID and gender. A separate analysis is done for each unique combination of sort variable values.

ID: Up to 5 categorical variable(s) identifying individual data profiles, such as subject ID and treatment in a crossover study. Only available in Population modeling.

Use the **Rebuild** button to reset the internal worksheet to its default state.

Note: Using an internal worksheet requires input for all values of the sort variables since missed covariate values are not interpreted properly.

Structure tab

The Structure tab is used to set up the structural level of a model and it is available to both built-in and graphic modes. When selections are made in this tab, the PML that defines the model changes. The changes to the PML can be viewed in the Model Text tab.

The following discussion of the Structure tab pertains only to built-in models. The description of this tab for graphic models can be found in the "Graphical model interface" section.

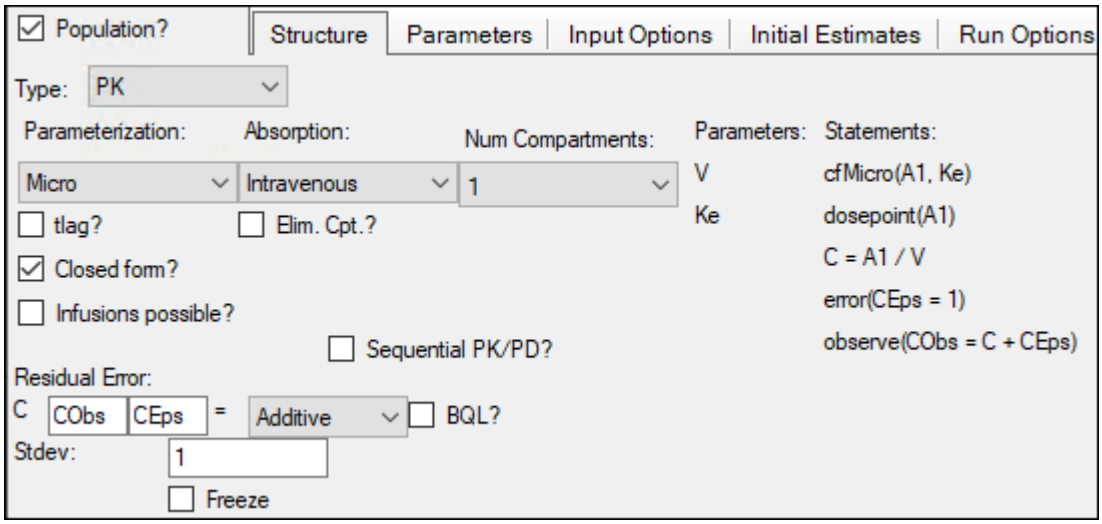


Figure 4-1. Structure model options tab set up for a PK intravenous population model

- In the **Type** menu, select the type of Maximum likelihood model:

  PK: See "PK model options".
  Emax: See "Emax model options".
  PK/Emax: The options for the PK and Emax portions of the model are the same as those listed under "PK model options" and "Emax model options", respectively. The other options are listed under "PK/PD model options".
  PK/Indirect: See "PK/Indirect model options" for Indirect model options. The options for the PK portion of the model are the same as those listed under "PK model options". The other options are listed under "PK/PD model options".
  Linear: See "Linear model options".
  PK/Linear: The options for the PK and Linear portions of the model are the same as those listed under "PK model options" and "Linear model options", respectively. The other options are listed under "PK/PD model options".

Caution: Changing the structural model after random effects have been set up can result in reordering of the random effects. Caution is advised in double-checking the random effects entries after a model is changed.

Structure tab options change depending on the type of model selected.

PK model options

- In the **Parameterization** menu, select the type of parameterization to use in the model. Each parameterization option changes the parameters and statements listed in the Structural tab.
  - **Micro**: Model is expressed as differential equations having mass transfer rate constants associated with a compartmental model. Adds the mass transfer rate parameters, such as Ke, the rate of elimination, and writes the derivatives in terms of compartmental masses.

- **Clearance**: Model is expressed as differential equations having clearances associated with a compartmental model. Adds the inter-compartmental clearance parameters, such as Cl, the clearance rate. If **Clearance** is selected, the **Saturating** checkbox becomes available. Check the **Saturating** checkbox to convert the model to a saturable elimination (Michaelis-Menten kinetics) model. A saturable model uses two parameters:
  Km: Concentration to achieve half of maximal metabolic rate.
  Vmax: The maximum metabolic rate.
- **Macro**: Model is expressed as a closed-form sum of exponentials. This option directly models ***concentration*** in the central compartment. Primary parameters are the sum of exponential terms that model concentration in the central compartment. Volume is not in the model but can be derived as a secondary parameter. Since Macro models use concentration, an assumption must be made about the size of a reference initial dose, called a "stripping dose". The user can enter the dose (A1) as well as a stripping dose (A1Strip). If no stripping dose is mapped on the Main or Dosing panels, the stripping dose is assumed equal to the dose.

  A note about stripping dose: The Macro model has a sum of exponentials that are fitted to the original observed value of C, using some original dose. If the model is again fitted against data obtained at another time with a higher or lower dose, then the question is how to make the model predict higher or lower values of C. This is done by multiplying the model's predicted value of C by a ratio of the current dose and the original dose. The value used for the original dose is called the "stripping dose", and it allows the model to be used with dose values and sequences different from the original.
- **Macro1**: Model is expressed as a closed-form sum of exponentials. This option models the ***amount*** in the central compartment (A) and Volume is a primary parameter. Primary parameters are the sum of exponential terms plus the parameter Volume. The amount in the central compartment is modeled as a sum of exponentials and then the Volume parameter is used to convert that amount to a concentration. Note that, because both macro options are closed-form models, the **Closed form?** option is removed.

- Turn on the **tlag** checkbox to add a time delay parameter to the model.

  Adding a time lag assumes that there is a fixed amount of time between drug delivery (dose time) and when the drug is introduced into the blood.
- Turn on the **Elim. Cpt.** checkbox if urine is being analyzed or data is available for any type of elimination compartment.

  Selecting this option adds a differential equation for A0, the amount of a drug excreted from the body. An elimination compartment is added to the model and included in the model code as a `urinecpt` statement. Basically, the `urinecpt` statement is like the `deriv` statement, except in a steady-state dosing situation, where `urinecpt` is ignored.

Note: Adding an elimination compartment to a Clearance model removes the **Closed form?** option and adds the **Fe.?** option for toggling the inclusion of the fraction excreted parameter.

- Check the **Closed form?** checkbox to convert the model from a differential equation model to a closed-form algebraic model.

  The closed-form model runs faster, but has some disadvantages:
  - There can only be one observed variable, such as central compartment amount or concentration.
  - The differential system being converted must be linear (no nonlinear kinetics).

    - Any covariates used in the model must be constant for the duration of the observations. For example, no change in subject's weight as samples are collected.
- Check the **Infusions possible?** checkbox if the model uses a constant rate of drug delivery.

  Infusions can be selected for intravenous and extravascular input. Selecting this option adds an extra context for the rate of drug delivery to the Main Mappings panel. For intravenous absorption, the new mapping context is **A1 Rate**. For extravascular absorption, the mapping context is **Aa Rate**. A **Duration?** checkbox is also added in the Structure tab. Checking this box causes the context **A1** or **Aa Rate** to change to **A1** or **Aa Duration**.
- In the **Absorption** menu, select the method of drug delivery.

  Intravenous: The drug is introduced directly into the blood.
  Extravascular: The drug is introduced indirectly into the blood.

  Selecting the **Extravascular** option adds an extra parameter, **Ka**, which is the rate of absorption and adds an extra checkbox that allows users to set the rate of absorption equal to the rate of elimination.
- Check the **Ka = Ke** checkbox (available when **Micro** is chosen under the **Parameterization** menu) to set the absorption rate equal to the elimination rate (the Ke parameter is removed from the model).
- In the **Num Compartments** menu, select the number of compartments.

  The default is one central compartment, and up to two peripheral compartments can be added. For each peripheral compartment added, two parameters are added to the model to account for the flow between the compartments. If macro parameterization is selected, the **Num Compartments** menu changes to the **Num Terms** menu.
- In the **Num Terms** menu, select the number of exponential terms to use in the model.

  The following lists the parameters that are added to the model for each extra compartment or term that is selected.

  1 Comp./Term: V, Ke (Micro); V, Cl (Clearance); A, Alpha (Macro); V, Alpha (Macro1)
  2 Comp./Terms: K12, K21 (Micro); V2, Cl2 (Clearance); B, Beta (Macro); B, Beta (Macro1)
  3 Comp./Terms: K13, K31 (Micro); V3, Cl3 (Clearance); C, Gamma (Macro); C, Gamma (Macro1)
- Select the **Sequential PK/PD?** checkbox if the PK model is part of a PK/PD model that is being fitted sequentially. This will freeze the PK portion of the model and turn its random effects into covariates. See “Sequential PK/PD population model fitting” for more information.
- In the **Residual Error** fields, define the equation used to determine the error model.

  C in the residual error model represents the central compartment. Changes made in the Residual Error field are shown in the **observe** statement.
- In the first field, type the name of the observed quantity variable or use the default (**CObs**).
- In the second field, type the name of the epsilon variable or use the default name (**CEps**).

  The epsilon variable represents a normal error with standard deviation as specified in the **Stdev** field.
- In the **error model type** menu, select the type of error model.

  The **Additive** error model assumes error magnitudes are constant, regardless of concentration. The error bars in the residual plots are set to CEps.
  The **Multiplicative** error model assumes error magnitudes are proportional to concentration. The

error bars are scaled by C*CEps.
The **Power** error model assumes the error magnitude is proportional to the concentration raised to the given power (i.e., CObs = C + CEps * C^p, where p is the value entered in the **Power** field).

- If **Add+Mult** is selected, type a name for the parameter in the **Mult Stdev** field or accept the default name.
- If **Mix Ratio** is selected, type a name for the mixed ratio parameter in the **mix Ratio** field or accept the default name.
- If **Power** is selected, type a value for the exponent in the **Power** field.
- If **Custom** is selected, type a custom error model definition in the **Defn** field.

For PK models, **Multiplicative** followed by **Power** are the preferred error models over **Additive**. This is because PK model types usually have concentrations spanning several orders of magnitude and, on a log scale, **Additive** has large errors at low concentrations.

For PD model types, with effect ranges usually less than an order of magnitude, **Additive** is the first choice.

- In the **Stdev** field, type a value for the standard deviation or accept the default.

  The residual error cannot be less than 0.001 for all engines except Naive Pooled and QRPEM.

- Select the **BQL?** checkbox if the dataset contains BQL values for the observation data. (The **BQL?** option is available for Emax models as well.) When checked, the engine automatically reverts to the Laplacian method.

  A resulting worksheet of a BQL object with a censored column **CObsBQL** can be used as the input for Maximum Likelihood Models with the option **BQL?**. If **BQL?** is selected, a column can be mapped to the **CObsBQL** context in the Main Mappings panel. This column can contain two categories of values: non-zero number (censored) or zero/blank (non-censored). A concentration value marked as censored (**CObsBQL**<>0 and it is not empty) means that the true value of the observation is unknown, but it is not greater than the observed value (e.g., LLOQ, which is provided in the CObs cell for that row) and then the cumulative distribution function for the normally distributed error is used to calculate the likelihood. (The likelihood is the probability of falling into the interval between minus infinity and LLOQ, where LLOQ is given the value of the CObs or EObs column on that row.) If a concentration value is flagged as non-censored, then the probability density function is used to calculate the likelihood.

  The context name **CObsBQL** changes based on what is typed in the observed quantity variable field. For example, instead of CObsBQL it could be ConcBQL.

Note: Maximum Likelihood Models with censored data (**BQL?** option) use the log of the probabilities between 0 and the censored number in the log likelihoods. If the censoring numbers are very small, the loglikelihood might overflow, resulting in a Fortran error. This seems to be more often the case when using multiplicative error models. If the error occurs, try increasing the BQL value if possible or change error types.

Note: For a quick and easy way to create an observation column and its associated BQL flag column, use the Phoenix BQL object.

- Turn on the **Use Static LLOQ?** checkbox to enter a numeric value of LLOQ (>0).

  In the event that **CObsBQL** is not mapped to a column in the dataset, then the static value of LLOQ will be used, so any observed value less than or equal to that LLOQ value is treated as

censored. Only turn on the **Use Static LLOQ?** box to specify a static LLOQ value. However, even if a value is specified, if the CObsBQL column (or other column with a BQL flag) is mapped, the value in the observation column will be used as the LLOQ and will override the static LLOQ.

- Check the **Freeze** checkbox to freeze the standard deviation to the value shown in the **Stdev** field and prevent estimation of this part of the model.
- Click **Set WNL Model** to run Least-Squares Regression model structures using Phoenix NLME.
  - In the first unlabeled menu, select one of the 19 PK models, or one of the four Michaelis-Menten models, or one of the 19 PK/PD simultaneous link models. The PK/PD simultaneous link models are labeled 401 to 419 and link PK models 1 through 19 with PD model 105 (e.g., PK/PD model 407 is a linked model of PK model 7 with PD model 105). For more on PK/PD simultaneous link models, see “Differential equations in NLME”.
  - If the model specified in the first unlabeled menu is ***not*** a macro-parameterization model, the **CL/V** checkbox is made available. Check this box to add clearance and volume parameters to the model.

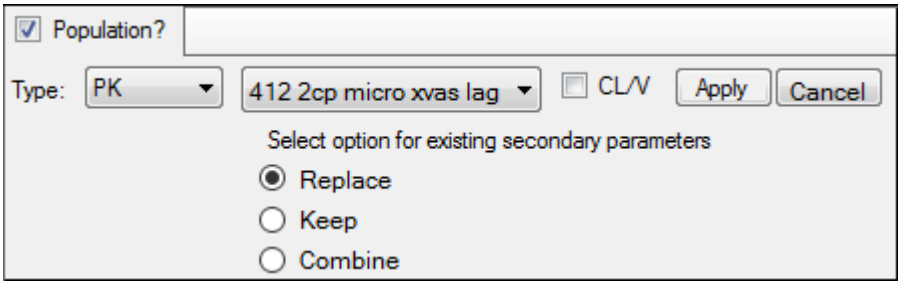


Figure 4-2. Least-squares regression model setup with additional parameters option

  - If a linked model is specified in the first unlabeled menu, a second unlabeled menu is made available. Select one of eight PD models, or one of four Indirect Response models to link.

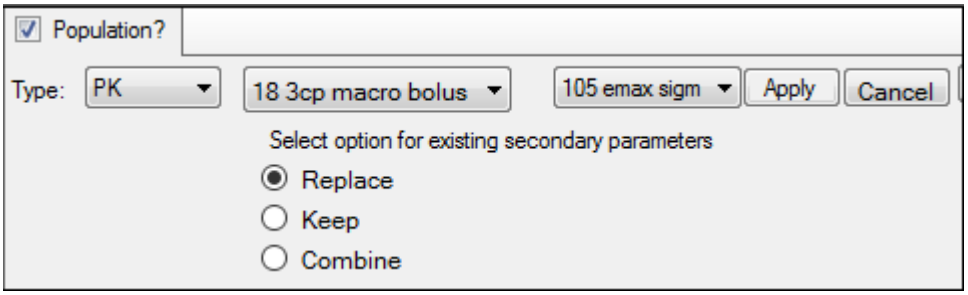


Figure 4-3. Least-squares regression linked model setup

  - Each Least-Squares Regression model has a set of secondary parameters that available for loading when that model is selected. Use the radio buttons to indicate what should be done with secondary parameters that are already defined in the interface when the model is loaded. Choose to **Replace** currently defined parameters with those and load the ones from the model, **Keep** currently defined parameters and ignore those from the selected model, or **Combine** them by keeping the existing parameters and adding those from the selected model to the list.
  - Click **Apply** to apply the selected model or models or click **Cancel** to exit the model selection menus without applying any changes.

Additional details on residual error models

The **Log-additive option** corresponds to a form like $C*exp(epsilon)$. If the **Log-additive** error model is specified, and if there is only one error model, such as one observe statement, then the predictions and observations are log-transformed and are fit in that space. This is because the error

model becomes additive in log-space, which allows for higher performance and accuracy. This affects all the plots results and residuals, because they are in log-space. The simulation tables are transformed back so they are not in log-space.

Because the logs of zero or negative numbers are not allowed, they are truncated to a value which is ¼ (0.25) of the smallest positive observation value. If the model is **Log-additive**, but the conditions have not been met for log-transforming to take place, the model behaves the same as **Multiplicative**.

The residual model that is displayed in the model text is:

$$\text{observe}(CObs = C * \exp(CEps))$$

or

$$\text{observe}(CObs = \exp(\log(C) + CEps))$$

The engines implement the model by log-transforming both sides and see if the derivative of the right-hand side with respect to CEps is 1. If so, and if there is only one observe statement, then it does log-transformation.

$$\text{observe}(\log(CObs) = \log(C) + CEps)$$

This check is accomplished by examining the text of the model, so it can be applied to models other than built-in models.

The main advantages of **Log-additive** are that the engines, particularly the Lindstrom-Bates FOCE engine, can run faster when a simple additive error model is used, and the FOCE approximation can be more accurate.

Since the modeling engines in NLME can only handle a single error variable (or epsilon), and some error models are best specified as having an additive component and a multiplicative component, some complexities are needed. The **Mix Ratio** uses the following formula:

$$C + eps * (1 + C * mixRatio)$$

where *mixRatio* is a fixed effect and is understood to be the multiplicative sigma (i.e., standard deviation of the multiplicative error variable) divided by additive sigma (i.e., standard deviation of the additive error model).

Another way to specify a mixed error model having a fixed effect, but with the fixed effect signifying the multiplicative sigma, rather than the ratio of multiplicative to additive sigma is **Add+Multi**. It makes use of a built-in function called "sigma()" that can only be used in this context, and its value is the current estimate of the standard deviation of eps. The formula is:

$$C + eps * \text{sqrt}(1 + (C * multStdev / sigma())\text{^}2)$$

where *multStdev* is the multiplicative sigma.

So when this error model is used, the additive sigma is called *stdev*, and the multiplicative sigma is called *multStdev*. Since *multStdev* is a fixed effect, its name can be changed as desire.

To justify the above formula, look at the variance. Suppose the additive standard deviation is called *sigma1*, the multiplicative standard deviation is called *sigma2*, and suppose the corresponding epsilons *eps1* and *eps2* are drawn from a unit normal distribution. Then the formula would be:

$$C + eps1 * sigma1 + C * eps2 * sigma2$$

The variance of this is the sum of the variances from each term, or:

$$sigma1^2 + (C * sigma2)^2$$

Now let r be the ratio: $r=sigma2/sigma1$, then the variance is:

$$sigma1^2+(C*r*sigma1)^2$$
$$\text{or } sigma1^2 *(1+(C*r)^2)$$

which is the variance of:

$$C+eps*\text{sqrt}(1+(C*r)^2)$$

Then replace r with $sigma2/sigma1$ to obtain:

$$C+eps*\text{sqrt}(1+(C*sigma2/sigma1)^2)$$

where $sigma1$ is represented by the $sigma()$ function, and $sigma2$ is represented by $multStdev$.

So, by choosing the option **Add+Mult**, this formula will be used to estimate both $stdev$ (the additive standard deviation) and $multStdev$ (the multiplicative standard deviation).

Emax model options

- Check the **Baseline** checkbox if the model has a baseline response.
  A new parameter, E0, is added to the baseline response and the **Fractional** checkbox is made available.
- Check the **Fractional** checkbox if the Emax model is fractional.
  Selecting this option modifies the E0 equation statement.
- Check the **Inhibitory** checkbox if the Emax model is inhibitory.
  The structural parameters EC50 and Emax change to IC50 (concentration producing 50% of maximal inhibition) and E0 (baseline effect).
- Check the **Sigmoid** checkbox if the Emax model is sigmoidal.
  A shape parameter, Gam, is added.
- In the **Residual Error** field, define the equation used to determine the error model.
  - **E** in the residual error model represents the effect observation.
  - Changes made in the Residual Error field are shown in the **observe** statement.
- In the first field, type the name of the observed effect variable or accept the default name of **EObs**.
- In the second field, type the name of the epsilon variable or accept the default name of EEps.
- In the **error model type** menu, select the type of error model. Refer to “Additional details on residual error models” in the PK model section for more information.
- Check the **BQL?** checkbox if the dataset contains BQL values for the observation data. Refer to the **BQL?** description in the PK model section for more information.
- In the **Stdev** field, type a value for the standard deviation.
- Check the **Freeze** checkbox to freeze the standard deviation to the value shown in the **Stdev** field.

Linear model options

- In the **Linear** menu, select the linear model type:

  E = Alpha, a constant model, where $y(x)=constant$.
  E = Alpha + Beta * C, a linear model, where $y(x)=intercept+slope*x$.
  E = Alpha + Beta * C + Gamma * C^2, a quadratic model, where $y(x)=A0+A1*x+A2*x\text{^}2$.

  In the above equations, y is the dependent variable and x is the independent variable.

- The Residual Error model options for the Linear model are the same as those for the Emax model options.

PK/PD model options

PK/PD models can be specified by selecting a PK/Emax, PK/Indirect, or a PK/Linear model from the **Type** menu or by setting a PK/PD link model as discussed above.

When a PK/Emax, PK/Indirect, or a PK/Linear model is selected, there are two observation columns displayed in the mappings panel: one for concentration observations; and one for effect observations.

The following options are specific to PK/PD models

- Check the **Effect Cpt?** checkbox to add an effect compartment to the model.
- Check the **Freeze PK?** checkbox to freeze the PK parameters by removing the PK observation from the model. Only the parameters for the PD model are estimated.
- Check the **Sequential PK/PD?** checkbox to estimate the PK and PD parameters sequentially.

  Checking the **Sequential PK/PD?** checkbox freezes the PK portion of the model and turns its random effects into covariates to be imported from the first model, so that plasma concentration will be predicted identically to what it was predicted as in the first model. It also removes the PK observation from the model, since the PK portion of the model is not being fit. See "Sequential PK/PD population model fitting" for more information.

To include a hysteresis plot, on the Run Options tab, add a Table with Times and Variables C, E, CObs, EObs, Ce for built-in models (or use the specified Variable names from graphical or textual models), and then create the desired plot from the Table output."

PK/Indirect model options

The following options are for the Indirect portion of the PK/Indirect model. Options for the PK portion of the model are the same as those listed under "PK model options".

- In the **Indirect** menu, select the indirect model type:

  Stim. Limited (limited stimulation of input)
  Stim. Infinite (infinite stimulation of input)
  Inhib. Limited (limited inhibition of input)
  Inhib. Inverse (inverse inhibition of input)
  Stim. Linear (linear stimulation of input)
  Stim. Log Linear (logarithmic and linear stimulation of input)

  Use the Indirect menu options to select a model in which the response formation (build-up) or degradation (loss) is stimulated or inhibited by increased concentrations. The default response setting is the build-up of the response, or the production of the response.

- Click **Build-up** to change the response setting to a **Loss**, or fractional turnover rate, of response.

  This constructs a model where the degradation of response is concentration dependent. For

example, Loss statement with limited stimulation:
deriv(E = Kin – Kout(1+Emax*C/(C+EC50))*E)*

- Click **Loss** to change the response setting back to **Build-up** of the response.

  This constructs a model where the formation of response is concentration dependent. For example, Buildup statement with limited stimulation:
  deriv(E = Kin(1+Emax*C/(C+EC50)) – Kout*E)*
- Click **no Exponent/Exponent** to toggle between adding and removing an exponent to/from the concentration dependent response build-up/loss term. For example, build-up with limited stimulation and exponent.
  deriv(E = Kin(1+Emax*C^gam/(C^gam+EC50^gam)) – Kout*E)*
- The Residual Error model options for the PK model and the Indirect model are the same as the PK model options and Emax model options.

Sequential PK/PD population model fitting

Generally, fitting a PK/PD model with population data is best done with a combined model, where the input data contains both PK observations (plasma concentration) and PD observations (effect). However, if it is desired to fit the model sequentially, for reasons of performance or modeling stability, that can also be done. This is done by creating two models, one for the PK model alone, and one for PK/PD, where the PK part of the second model imports data from the results of the first model.

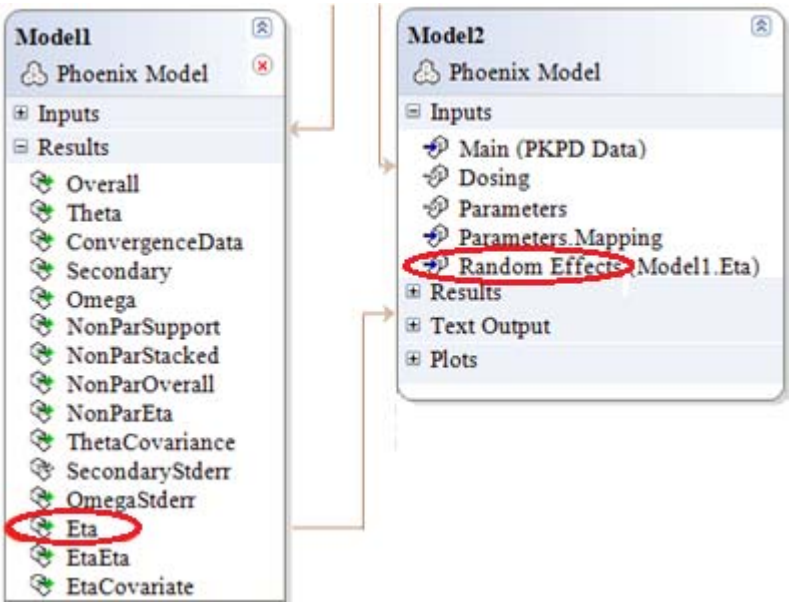
- Create the PK-only model and fit it against the combined input data set, ignoring the effect observations.
  - The **Sort Input?** option in the Run Options tab must be checked.
  - Refine the model until it is considered satisfactory.
  - Be sure to select **Accept All Fixed + Random** under the **Parameters > Fixed Effects** tab, so that the estimated values are transferred to the initial values.
- Copy the PK-only model to a second model and convert it to a PK/Emax model (Structure tab).

  This ensures that the PK portion of the second model is identical to the PK-only model. It must be identical in structure, have the same structural, fixed effect, and random effect parameters, and initial estimates. It must also have the identical main input datasets, mapped in the identical way.
- Check the **Sequential PK/PD?** option on the Structure tab.

  This will freeze the PK portion of the model, and turn its random effects into covariates to be imported from the first model, so that plasma concentration will be predicted identically to what it was predicted as in the first model. It also removes the PK observation from the model, since the PK portion of the model is not being fit.

Note: For built-in or graphical models, if a fixed effect is frozen, then the corresponding random effect is removed from the model. This may cause confusion because the user interface may appear as if the random effects corresponding to the frozen fixed effects are supported. See “User interface settings and the sequential PK/PD” for more information.

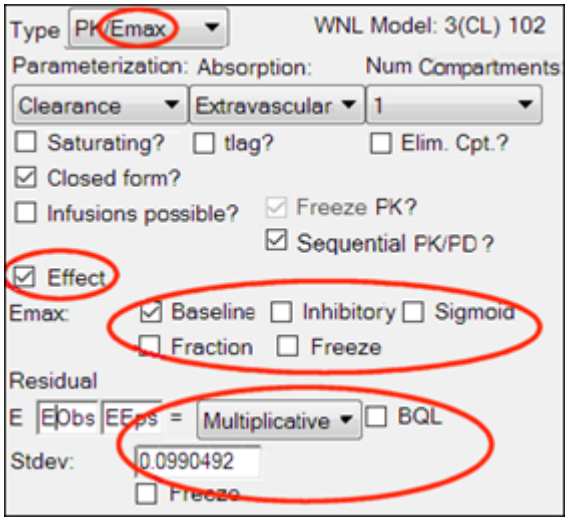
- Map the **Eta** output of the first model to the **Random Effects** input of the second model.



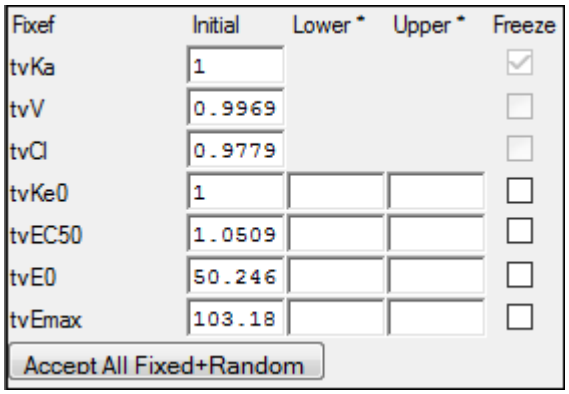
For example, if **ID** is mapped to the **ID** in the Main Mappings panel and Eta worksheets contain two random effects, **nV** and **nCl**, then make sure the following mapping is completed correctly:

Scenario is mapped to **None**.
ID is mapped to **ID**.
nV is mapped to **nV**.
nCl is mapped to **nCl**.

- Set up the PD portion of the model.



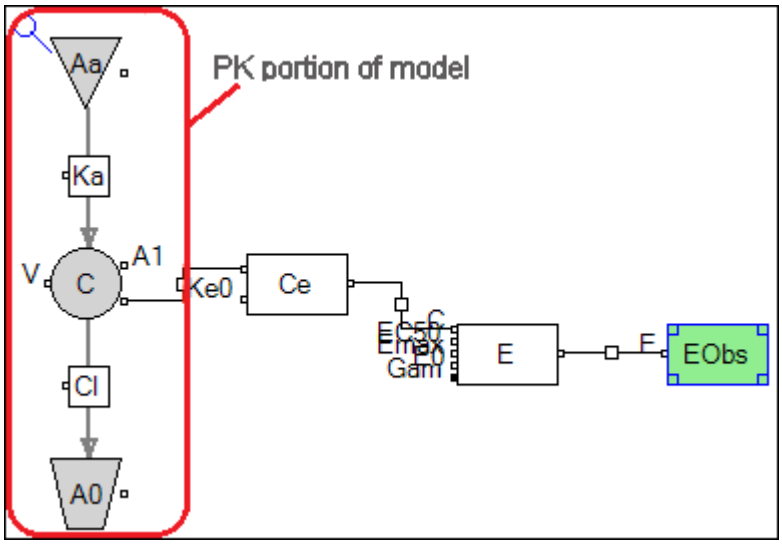
Look at the Fixed Effects tab. When the model is run, it will only fit the fixed and random effects for the PD portion of the model. The PK parameters are not estimated because they come from the first model:



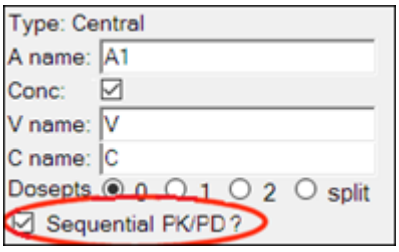
Look at the Run Options tab. All run modes can be used. Note that **Scenarios** and the two **Cov. Srch.** modes will only apply to covariate effects in the PD portion of the model. PK covariate effects must be unchanged from the first model. In the **Profile** mode, the only fixed effects available for profiling are the PD fixed effects.

To convert the second model to graphical, it is recommended to first set it up in built-in mode, and then use **Edit as Graphical**.

The resulting graphical model has no PK observation:



Every component of the PK model has a checkbox called **Sequential PK/PD**, which is checked.

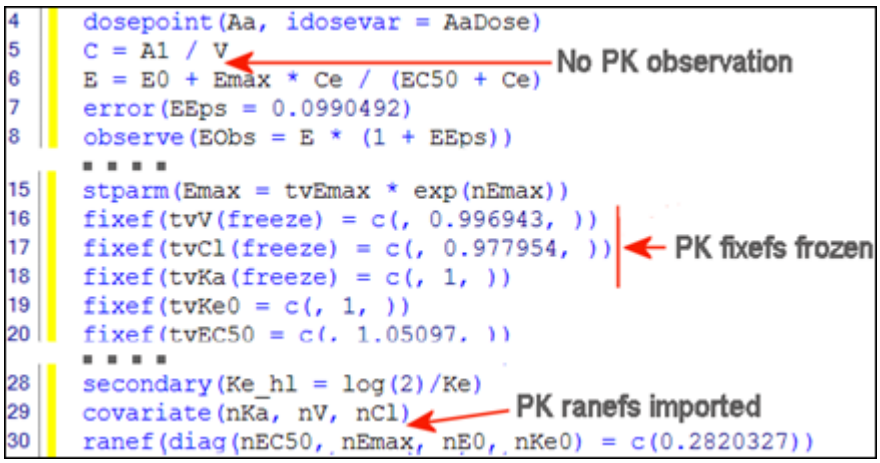


This checkbox causes the parameters associated with that compartment or flow to be treated as frozen, with imported random effects. (This is necessary because a graphical model is more modifiable than a built-in model. Since the user may make changes, they must tell NLME which components are PK components.) When the graphical model is generated from the built-in model, these options are automatically set.

The PD portion of the model can be modified as desired. The remaining usage is the same as that described for the built-in model.

To make a textual model equivalent to a built-in or graphical model, it is recommended to start with a built-in or graphical set up and then select **Edit as Textual**.

The text shows the result:



Now the PD portion may be modified as desired, and executed as in the built-in or graphical models.

User interface settings and the sequential PK/PD

The user interface settings may appear to allow random effects on fixed effects, even though this situation is not supported. For example, if random effects for PK parameters are selected first, then the corresponding fixed effects are frozen, the user interface will appear as if random effects on fixed effects are allowed, since both checkboxes are selected, but the model text will not have these random effects.

In addition, the interface allows selection of **Sequential PK/PD?** without having any random effects in the model. In this case, the model will execute without error, even though it is not a true sequential PK/PD model, since random effects from the PK model are not used. The **Ran** checkboxes for the corresponding fixed effects need to be selected prior to selecting the **Sequential PK/PD?** checkbox in order to run as a true sequential PK/PD model.

If a user wants to run a sequential PK/PD model and, due to loading an existing project or due to freezing fixed effects prior to checking **Sequential PK/PD?**, the interface is in a state where **Sequential PK/PD** is selected, and the **Freeze PK?** checkbox or the **Freeze** checkboxes for the fixed effects are selected and disabled (thereby removing the random effects from the model text), the user must:

1. Unselect **Sequential PK/PD**
2. Unselect the **Freeze PK?** checkbox or the **Freeze** checkboxes for the fixed effects
3. Select the **Ran** checkboxes for the corresponding fixed effects (if not already selected)
4. Re-select **Sequential PK/PD?**
5. Map the random effects on the Random Effects mappings panel on the Setup tab

Parameters tab

The Parameters tab contains six sub-tabs that allow users to modify the structural parameters, specify values for the fixed and random effects, and add covariates.

Covariate effects can be specified in population modeling as well as in individual modeling, however, care must be taken not to over-parameterize the individual model. For example, suppose body weight W affects volume V. Then the model for V might be V=tvV+W*dVdW. In this case, if W is constant for the individual, the model is over-parameterized, because tvV and $dVdW$ are redundant. However, if W is time-varying, the model is not over-parameterized. Covariates can also be useful if the data are pooled, or all subjects are modeled together.

Other covariates can be included in the individual model, even though they may not affect any structural parameters, because they may appear in secondary parameters, such as AUC.

Columns mapped as covariates (including categorical or occasion covariates) can ***only*** be numeric. The user can associate a label to the numeric categorical covariate, for example, and the label will be displayed in graphs even if the underlying data is numeric.

Structural sub-tab

The Structural tab lists all the structural parameters used in the model. The listed parameters change depending on the selections made in the Structure tab.

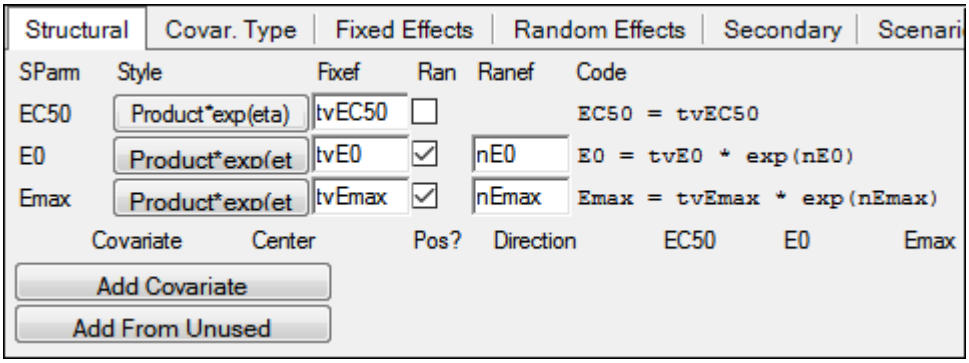


Figure 5-1. Structural Parameters tab for Emax model

Every selection made in the structural tab changes the code for the modified structural parameter. These code changes are displayed in the Model Text tab.

- Click the buttons below Style multiple times to toggle through the different style options for each structural parameter: **Sum+eta**, **Product*exp(eta)**, **Sum*exp(eta)**, **exp(Sum+eta)**, **ilogit(Sum+eta)**.

  The most common recommended form is Product*exp(eta) for positive-only parameters like V, Cl, or various Ks. For parameters like E0 or Emax, which can be positive or negative, Sum+eta is the preferred choice, For parameters that are constrained to fall between zero and one, ilogit(Sum+eta) is a useful choice.

  If there are no covariates, Product*exp(eta) and Sum*exp(eta) yield nearly identical expressions

in the model code. The differences between the two are seen when there are covariates and they come into the equation either through multiplication or addition. For example, here is what Product*exp(eta) provides in the presence of covariate effects, where the user has chosen V and Cl on Gender, wgt, and apgr:

```
stparam(V=tvV
    *(wgt/mean(wgt))^dVdwgt
    *(apgr/median(apgr))^dVdapgr
    *exp(dVdGender1*(Gender==1))
    *exp(dVdGender2*(Gender==2))
    *exp(nV)
    )
stparam(Cl=tvCl
    *(wgt/mean(wgt))^dCldwgt
    *(apgr/median(apgr))^dCldapgr
    *exp(dCldGender1*(Gender==1))
    *exp(dCldGender2*(Gender==2))
    *exp(nCl)
    )
```

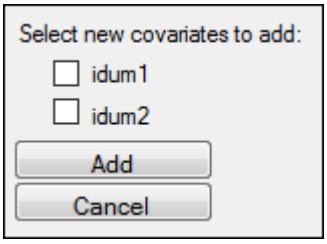
Here is what Sum*exp(eta) gives you in the presence of covariate effects:

```
stparm(V-(tvV
    +(wgt-mean(wgt))*dVdwgt
    +(apgr-median(apgr))*dVdapgr
    +(Gender==1)*dVdGender1
    +(Gender==2)*dVdGender2
    )
    *exp(nV))
stparm(Cl=(tvCl
    +(wgt-mean(wgt))*dCldwgt
    +(apgr-median(apgr))*dCldapgr
    +(Gender==1)*dCldGender1
    +(Gender==2)*dCldGender2
    )
    *exp(nCl))
```

- In the **Fixef** field type a name for the fixed effect or use the default (`tv` (typical value)+the parameter name (`tvV`, `tvKe`, etc.)).
- Check the **Ran** checkbox beside each parameter to add a random effect to the structural parameter (the parameter is added to the Random Effects tab).
- In the **Ranef** field, type a name for the random effect or use the default name (`n` (eta)+the parameter name (`nKa`, `nV`, `nKe`, etc.)).

The bottom section of the Parameters' Structural tab allows users to add covariates to the model structure.

- Click **Add Covariate** to add a covariate or click **Add From Unused** to add a covariate from the main input dataset.
  A list of all variables mapped to **None** in the Main Mappings panel is displayed.



- Turn on the checkbox beside each variable to add as a covariate.

- Click **Add** to add the variable(s) as covariates or click **Cancel** to exit the **Add From Unused Columns** menu without adding any covariates.

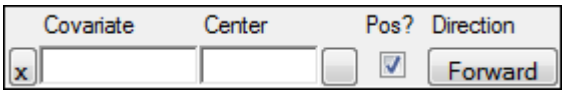


Figure 5-2. Covariate settings

To remove a covariate from the model, click the corresponding **X** button.

- In the **Covariate** field enter a name for the covariate or use the default.
- In the **Center** field, type the centering value for the covariate or click the button following the field to toggle between entering numeric value, using the **mean**, or the **median**.

  Only continuous covariates can have a center value. See "Covar. Type sub-tab" for instructions on selecting the covariate type.
- Clear the **Positive?** checkbox if the covariate values are ***not*** all positive.
- Click **Direction** to specify the method of curve fitting if the covariate value changes between observations for a subject.
- Click **Forward** multiple times to toggle through the curve fitting methods:

  Forward holds the first value between covariate observations.
  Interpolate linearly interpolates the covariate between covariate observations. Only available for Continuous covariate types.
  Backward holds the last value between covariate observations.

  Covariates can also be added based on available columns in the input source. See "Considerations when modeling with covariates" for additional information on covariate direction.
- Under each structural parameter, there is a button to indicate how a given covariate can effect that structural parameter. The default is **No** (no covariate effect). Click the button multiple times to toggle through the other two options: **Yes** and **1+**. The **1+** option is only available for Product*exp(eta) structural parameters, and is not the recommended choice.

Adding covariate effects to structural parameters

Users can add three types of covariate effects to structural parameters. They are continuous, categorical, and occasion. Each type has its own set of options, and affect the structural parameters and the model differently.

The structural parameters are displayed beside each covariate that is added.

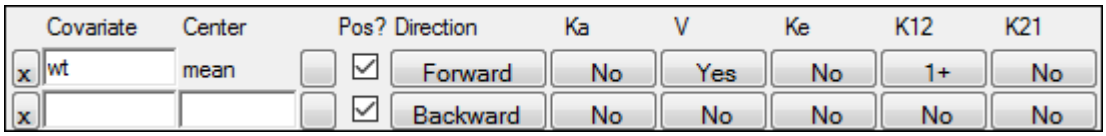


Figure 5-3. Covariate settings with structural parameters displayed

Each time a covariate effect is added, the code in the Structural tab and in the Model Text tab is modified.

Continuous covariate effects

Each parameter has a button that toggles between three values as it is clicked: **No** (the default), **Yes**, **+1**. The value shown on the button when the object is executed defines how covariate effects are added to structural parameters.

- **No** does not add covariate effects to the parameter.

- **Yes** adds covariate effects to that parameter by updating the code with an additional term

  For example, if the effects of the covariate *wgt* are added to the structural parameter V, a new fixed effect parameter is created called *dVdwgt* and the term *wt^dVdwgt* is added). *dVdwgt* is also added to the Fixed Effects tab, and users can enter initial, lower, and upper values for the fixed effects parameter, In this example, *dVdwgt* is the derivative of the parameter value with respect to weight. dV is the increment of volume divided by *dwgt*, the increment of weight.

- **+1** also adds covariate effects to the parameter by updating the code with an additional term (e.g., for the covariate *wgt* added to the structural parameter V, the term $1+wt*dVdwgt$ is added).

Each covariate effect added creates a new fixed effect in the Fixed Effect tab. The new fixed effect can be modified in the same way as any other fixed effect.

Categorical covariate effects

Each parameter has a button that toggles between three values as it is clicked. The value shown on the button when the object is executed defines how covariate effects are added to structural parameters.

Users ***cannot*** enter center values for categorical covariates.

No does not add covariate effects to the parameter.
Yes adds covariate effects to that parameter by updating the code with an additional term
+1 also adds covariate effects to the parameter by updating the code with an additional term.

Each covariate effect added creates a new fixed effect in the Fixed Effect tab. The new fixed effect can be modified in the same way as any other fixed effect.

Occasion covariate effects

The occasion covariate effect is used in a different way for variables. For example, the occasion could specify whether an observation was taken on a Monday or a Wednesday.

Each parameter has a button that toggles between two values as it is clicked. The value shown on the button when the object is executed defines how covariate effects are added to structural parameters.

No does not add covariate effects to the parameter.
Yes adds covariate effects to that parameter by updating the code with an additional term.

Adding an occasion covariate creates a copy of each selected structural parameter random effect in the Random Effects tab. For example, if V is a structural parameter, and an occasion covariate is added to it, then nV is added to the Random Effects tab, where n stands for eta, or random effect, and V stands for volume. If three occasion covariate effects are added for V, they are named nV, $nV2$, and $nV3$.

The new random effect can be modified in the same way as any other random effect.

Considerations when modeling with covariates

Covariates may have apparent incorrect covariate values being propagated (contrary to observed data), because of forward/backward/interpolate, for time-varying covariates. This raises several significant issues to consider when modeling:

- It is important to note that covariates have a direction of propagation that is forward in time, backward in time, or linearly interpolated.

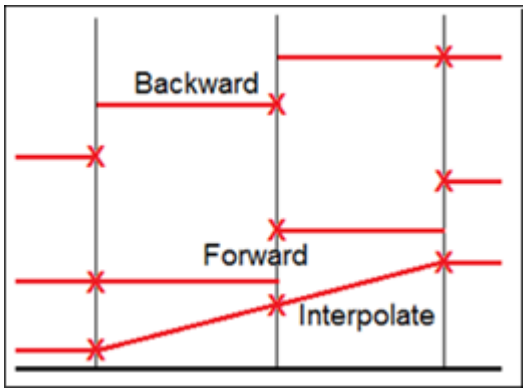


Figure 5-4. Illustration of propagation direction

The default is forward in time, to be somewhat consistent with other tools. (Refer to "Structure tab" for setting the direction.)

- The propagation occurs over records with missing covariate values, which is not consistent with some other tools. The user needs to be aware that if an observation has a missing covariate value the covariate will take on a propagated value, rather than a zero value.

Note: In situations where a covariate is missing because it has not been mapped, Phoenix NLME exits with an error message. If the covariate is mapped, but one or more subjects do not have a row of data for that covariate, Phoenix NLME also exits with an error message. For example, if one subject did not have the variable "gender" at all and the model includes "gender" as a covariate for V, for instance, then the full model will fail.

Note: There is no concept of a default value for a completely missing covariate, missing rows of data need to be resolved in the dataset prior to modeling.

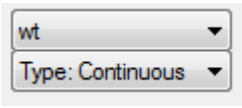
Regardless of covariate direction, the value of the covariate applies during all other dose or observation events occurring on the same input data row. For example, if the data looks like this:

```
time  CObs  age weight
…
  14   99.7   17     50
…
```

If there is any `doafter` code associated with observable CObs, the age and weight have the value 17 and 50 in that code, regardless of forward or backward covariate direction.

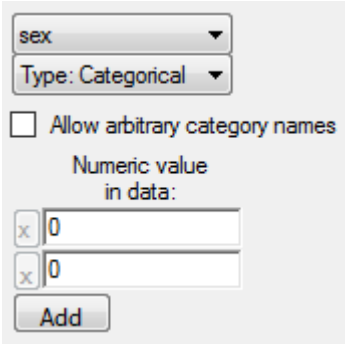
Covar. Type sub-tab

The Covariate Type tab allows users to specify covariate types. The default setting for each covariate is **Continuous**.



- In the top, unlabeled menu, select the covariate.
- In the **Type** menu, select the covariate type: **Continuous**, **Categorical**, **Occasion**.

If **Categorical** is selected, users can enter values for the categories. A minimum of two categories is required for categorical covariates.

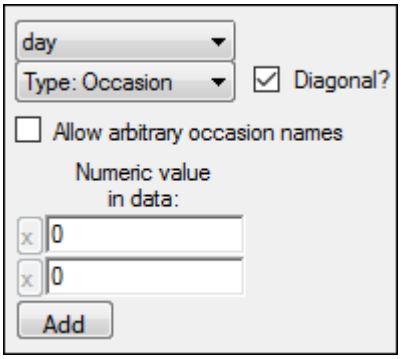


- In the **Numeric value in data** field, type a value for each category. It is typically best to use consecutive integers, starting at zero.
- If the **Allow arbitrary category names** checkbox is checked, the user is able to specify a name for each category and enter an associated value. The categorical values entered must be in the main input dataset.

Note that the actual text values that were in the dataset will appear in the Settings output and History to clarify the mappings that were set (e.g., `covr(Gender<-"TextGender"(Male=0, Female=2))`).

- To add another category, click **Add**.
- To remove a category, click the corresponding **X** button.

If **Occasion** is selected, users can enter values for the dosing or observation occasions. A minimum of two categories is required for interoccasional covariates (refer to "Setting up interoccasional covariates").



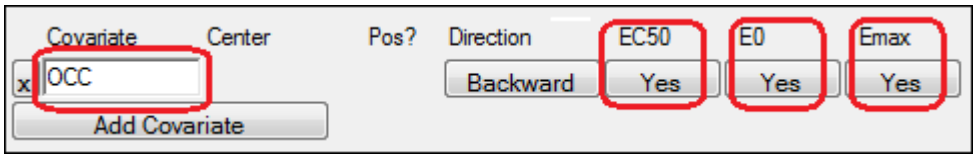
- In the **Numeric value in data** field, type a value for each occasion.
- Check the **Allow arbitrary occasion names** checkbox to specify a name for each category and enter an associated value. The occasion values entered must be in the main input dataset.
- To add another dosing or observation occasion, click **Add**.
- To remove a dosing or observation occasion, click the **X** button.
- Use the **Diagonal?** checkbox to set the interoccasion covariance to a diagonal structure (box is checked, this is the default) or block structure (box is unchecked).

Occasion covariates are always available but they are only used with population models. They are not used with the individual models that Phoenix processes. Using occasion covariates with individual models has no effect on the model or the output.

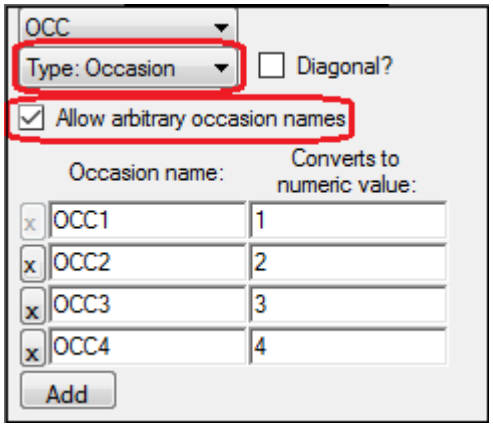
Setting up interoccasional covariates

If there is to be interoccasion variability (IOV) in a population model, there are several steps to follow.

- In the Parameters/Structural tab add a covariate and specify the variables it is to effect.



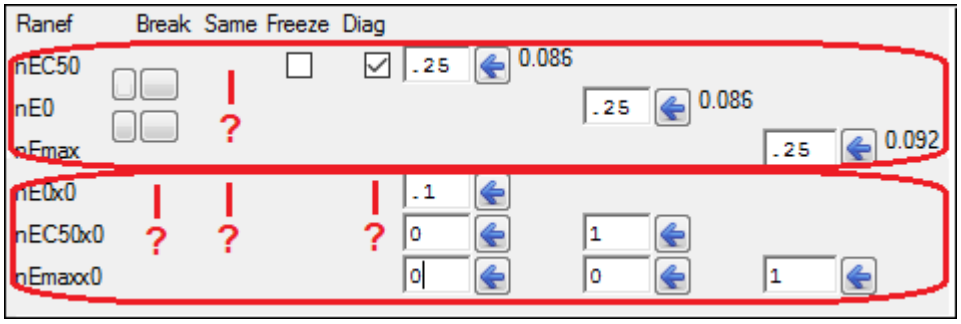
- Then, on the Covar. Type tab, set the type of the covariate to **Occasion**.



Note that, if **Allow arbitrary occasion names** is checked, two columns are presented in which arbitrary occasion names (including non-numeric) can be entered in the left column and corresponding numeric values on the right.

The checkbox labeled **Diagonal** is discussed a little later.

- Select the Random Effects tab.



In the upper box are simple random effects, one for each structural parameter that has randomness. The reason there is no checkbox under **Same** is because there is only one block, and that option only appears between blocks and only if they are the same shape. If **Same** is checked, it means that the lower block shares the same matrix as the upper block. In this case, the lower block is not displayed because its numbers cannot be edited.

The lower box has the first block of random effects for the occasion covariate effect. There are actually four blocks, one for each value of the occasion covariate, but only the first is shown, because the other three are all the same as the first. Note that there is no button for **Break**, and no checkbox for **Same**. That is because the IOV random effect structure is entirely specified by the choices made on the previous tabs, so they cannot be changed here. Also note that, in this case, the omega matrix for these three random effects is a lower-triangle block, not diagonal. It can be made diagonal by checking the **Diagonal** checkbox previously mentioned on the Covar. Type tab.

It is helpful to see what this does in the generated model text.

- OCC is declared to be a covariate:

```
covariate(OCC)
```

- The random effects are declared using the "same" notation:

```
ranef(block(nE0x0,nEC50x0,nEmaxx0)=c(0.1,0,1,0,0,1)
     , same(nE0x1,nEC50x1,nEmaxx1)
     , same(nE0x2,nEC50x2,nEmaxx2)
     , same(nE0x3,nEC50x3,nEmaxx3)
     )
```

  Note that there are four sets of three random effects each. The omega matrix for the first set is shared by the following three.

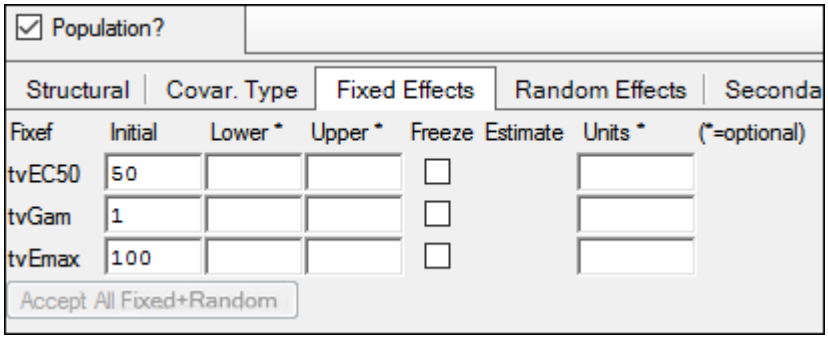
- The way these random effects appear in the model is that the value of the OCC covariate selects which random effect is active at any one time, like this:

```
stparm(EC50=tvEC50*exp(nEC50
     + nEC50x0*(OCC == 1)
     + nEC50x1*(OCC == 2)
     + nEC50x2*(OCC == 3)
     + nEC50x3*(OCC == 4)
     ))
```

(There are further statements for each of the structural parameters to be effected.) This differs from a typical categorical covariate effect in which the covariate selects a fixed effect, as opposed to a random effect.

Fixed Effects sub-tab

The Fixed Effects tab allows users to enter initial, lower bound, and upper bound values for the fixed effects. Every selection made in the Structural tab changes the code for the modified structural parameter. These code changes are displayed in the Model Text tab.



Entering lower and upper bound values for the fixed effects are optional. (See"Using upper and/or lower bounds" below for additional information.)

- In the **Initial** field, type an initial value for each fixed effect.
- In the **Lower** field, type a lower limit value, if needed.
- In the **Upper** field, type an upper limit value, if needed.
- Check the **Freeze** checkbox to fix the parameter estimates to the values entered in the initial, lower, and upper fields.

  The Estimate area is blank before a model is executed, but will automatically show parameter

estimates after the model is successfully executed. Users can choose to accept these estimates as the new initial estimates. (See “Parameters panel” for more information on estimate setup.)

- Click **Accept All Fixed+Random** to copy all new initial estimates values for all fixed effects, random effects, and the standard deviation.

The Estimate area only shows the results of the last model run, so when modeling individual subjects, only the estimates of the last subject are displayed.

- In the **Units** field(s), enter the desired unit of measurement for the fixed effect. When executed, the units are applied to the output. If the data has units, the fields are disabled. (See “Units labeling” for more information on handling of units.)

Note that if initial parameter estimates are entered in the Fixed Effects tab, then each sort level model will use the same initial estimates. On the other hand, if the user enters the initial parameter estimates using a worksheet that is mapped to the Parameters panel from the Setup list (either internal or external worksheet), then different initial estimates can be used per sort.

Use the Initial Estimates tab to visually determine a set of parameter estimates that approximate the data. See “Initial Estimates tab”.

Using upper and/or lower bounds

Only use the lower and upper bounds if the model converges on a solution that makes no physical sense, such as a negative rate constant or negative volume. If bounds are entered, the program automatically transforms the parameter to an unbounded space.

For a one-sided bound, a square root transformation is used. For a two-sided bound, a square root transformation is first applied to the original parameter, and then an arcsine transformation is applied to the resulting parameter. To avoid obtaining a negative value, a logarithm transformation is used for the standard deviation of the last residual error, with its lower bound automatically set to be 0.001. This applies to all the engines except for the QRPEM and Naïve-Pooled engines, where the lower bound is set to 0. Moreover, for the QRPEM engine, all bounds are ignored except the embedded lower bound for the last residual error.

It is worth pointing out that all transformations are done internally during the optimization phase and the transformed parameter never appears in any results reported to the user. The internal parameter in unbounded space is always transformed back to the original bounded space before results are reported.

In situations where **Multiplicative+Additive** residual error models are used or any custom error model involving the standard deviations of some residual errors, there is a possibility of a negative result being obtained for the standard deviation. In such situations, the absolute value of the standard deviation should be considered as the concluding result. To avoid this, the user needs to set a lower bound (e.g., 0) for the parameter.

Units labeling

Phoenix allows the user to label the model parameters (primary as well as secondary) with units. These units do not play a role in the model fitting and no conversion takes place.

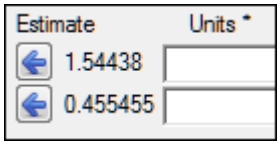
When an input dataset has the pertinent columns with units that define units for the model parameters (e.g., for a PK model without covariates: time, concentration and dose), then these units are carried to the output for fixed effects and secondary parameters. In this case, the user is not allowed to enter different units when specifying the model. The units for each column need to be part of the header for Phoenix to understand them as units and carry them forward to the model results. Note that as long as there is text in the unit location, it will be used as unit labels for the fixed effects and secondary parameters. Other Phoenix tools require that the units are valid (as denoted by being between paren-

theses) to perform unit transformations, but this is not a requirement for Phoenix NLME as these are merely used as labels.

Source Data

	xid	dose	time (hr)	yobs (mg)	wt (kg)
1	1	4.02	0	.	79.6
2	1	.	0	0.74	.
3	1	.	0.25	2.84	.
4	1	.	0.57	6.57	.

If the input dataset has no column units, then the user can optionally enter the units for each of the fixed effects as well as secondary parameters. These fields do not have any specific requirement as they are used as labels.

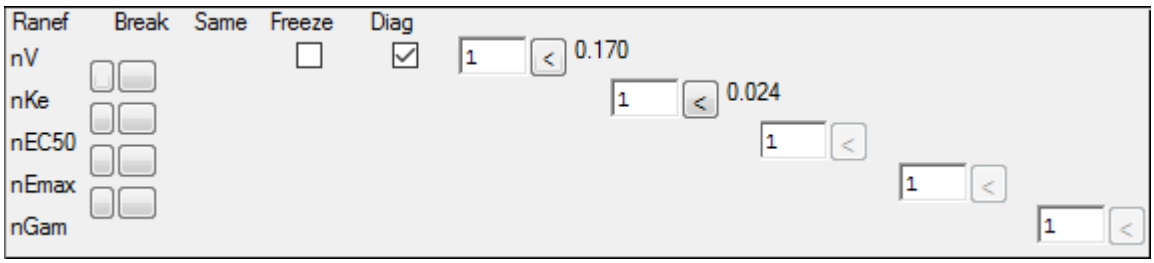


There are no requirements for the units as they are used as labels, but it is a good practice for the user to make sure that the units make sense within the dataset (e.g., the concentration mass unit is the same as the dose unit) and that the initial estimates are provided in the expected units. The Data Wizard can be used to do any unit transformation prior to modeling. See the "Data Wizard" description.

It is recommended that the input dataset be consistent with regard to units that define the model parameters. In other words, data values should preferably be in the same mass, time, volume, etc. units.

Random Effects sub-tab

The Random effects tab is only available for population models. Changes made to random effects are reflected in the Model Text tab.



- Click the **Swap random effects** button multiple times to toggle the positions of the random effects and customize the omega matrix structure.

  Random effects in the omega matrix can be swapped as many times as needed. For example, it is possible to move the first fixed effect to the bottom of the list and move the last fixed effect to the top of the list.

- Click the **Split or join the random effect blocks** button, shown under **Break**, multiple times to toggle splitting or joining the random effects and create multiple omega matrix structures.

  The **Same** checkbox is displayed if the random effect blocks are split more than once, or if split-

ting the blocks groups together two or more random effects separately from the others. At least two distinct blocks must be created before the **Same** checkbox can be used, and the blocks must be of the same size.

- Check the **Same** checkbox if the random effects block is the same as the previous one.

  Any selections made to the previous block are reflected in the ones marked as the same. Blocks that are marked as being the same have all their user interface options removed.

Note: When a covariate is declared an interoccasion covariate in the UI, the Random Effects tab provides an omega block for a single occasion. Therefore, the **Break** and **Same** boxes will not be available for this part of the omega matrix. A textual model would need to be used if a different behavior were desired.

- Check the **Freeze** checkbox to freeze the initial omega estimate values at their set values by removing the omega elements from the optimization routine.

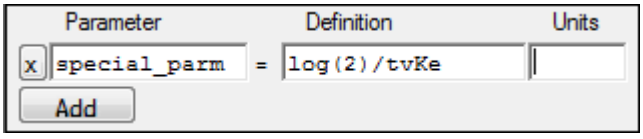
  The initial estimates are still used in the model, but Phoenix does not try to optimize the estimates and does not offer better estimates after a modeling run. In addition, the eta values are different in the output when the initial omega estimate are frozen.
- Use the **Diagonal?** checkbox to set the interoccasion covariance to a diagonal structure (box is checked, this is the default) or block structure (box is unchecked).

  After a model is run, and if the random effects are not frozen, then Phoenix automatically displays new omega estimates.
- Click the **Copy omega estimate to initial estimate** < button to copy the new estimate to the initial omega estimate field.

Secondary sub-tab

The Secondary tab allows users to add secondary parameters to the model.

- Click **Add** to add a secondary parameter.
- In the **Parameter** field, type a name for the secondary parameter.
- In the **Definition** field, type a definition for the secondary parameter.
  - Enter the right side of an equation that uses parameters available in the model.
  - The equation must be a function of fixed effects such as tvV and/or covariates. Refer to the "Supported Math Functions" and "Supported Special Functions" for a list functions that are supported.
  - Secondary parameters depending on categorical covariate effects do not work for Built-in or Graphical models. The user interface does not accept them. However, they do work for Textual models. For example, if "sex" is a categorical covariate having values 0 and 1, and it modifies column "V," then there is a fixed effect named "dVdsex1." This fixed effect is not recognized in the secondary parameter definition; however, it will work in a Textual model.
- In the **Units** field, enter the units for the parameter, if needed. When executed, the units are applied to the output. If the data has units, the field is disabled.



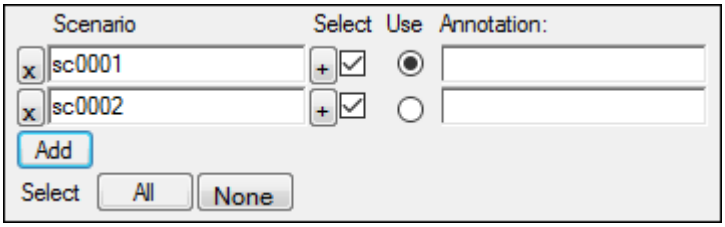
- Click the **X** button to remove a secondary parameter.

Note: Tmax can be defined as a secondary parameter by using the built-in function CalcTMax, e.g.:
`secondary(Tmax=CalcTMax(tvA,tvAlpha,tvB,tvBeta,tvC,tvGamma))`
If a Tlag variable is included in the dosepoint statement, the dosing is postponed by Tlag, so Tlag should be added into the secondary parameter definition for Tmax, e.g.:
`secondary(Tmax=CalcTMax(tvA,tvAlpha,tvB,tvBeta,tvC,tvGamma)+tvTlag)`

Scenarios sub-tab

The Scenarios tab is only available for population models. The Scenarios tab lists all covariates effects in the model. If there are no covariate effects in the model, then there is no need to add scenarios.

If the Covariate Search Stepwise or the Covariate Search Shotgun run options are selected in the Run Options tab, then Phoenix automatically creates scenarios during the modeling run, but only if covariate effects are used in the model. The new scenarios are added to the Scenarios tab.



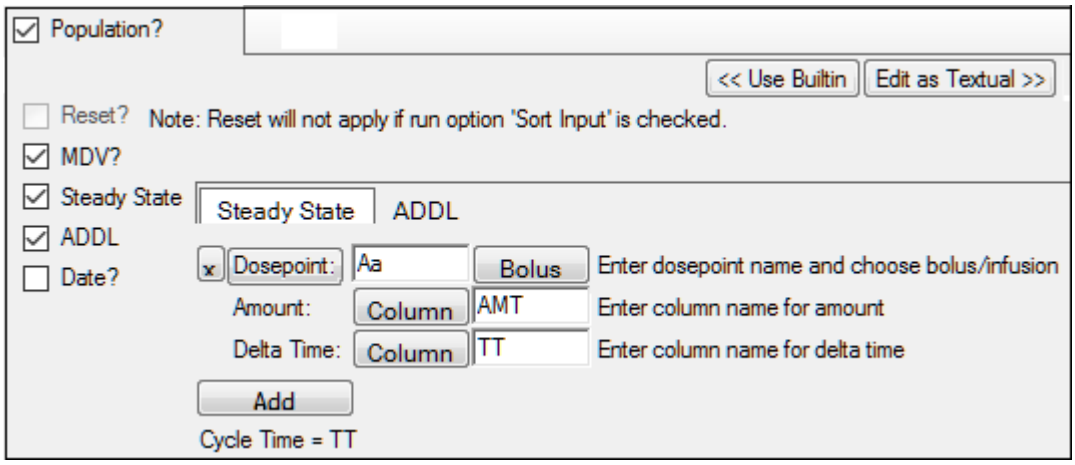
Note: Scenarios are run in the order they are listed on the Scenarios tab.

- Click **Add** to add a scenario to append a new scenario,
  Or
  Insert a new scenario between scenarios by clicking the **+** button of the preceding scenario.
- Accept the default scenario name or type a new one.
- Check the **Select** checkbox to include the scenario(s) in a modeling run.
- In the **Annotation** field, type a description of the scenario.
- Select the **Use** option button for the scenario to use when the Simple run mode is selected in the Run Options tab.

  In Simple run mode, only one scenario can be used during the modeling process. The **Use** option button determines which scenario is used during a simple run, no matter how many scenarios are selected.
- Click **All** to select all scenarios.
- Click **None** to clear all scenario selections.
- To run the scenario(s), select the Run Options tab and choose the **Scenarios** run mode.

Input Options tab

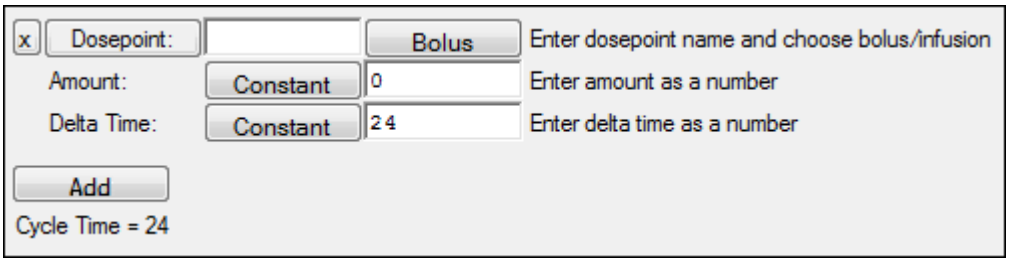
The Input Options tab allows users to add extra mapping contexts to the Main Mappings panel and convert fixed effect units.



- Clear the **Sort Input?** checkbox in the Run Options tab to make the **Reset?** option available.
- Check the **Reset?** checkbox if the main input dataset contains a reset column.

  This option sets the state variables of differential equations to zero and restarts any sequence statements in the model. The low value and high value fields for the Reset option allow a range of values to be used to trigger a reset. The default is to reset when a value exactly equal to one is found in the Reset column.
- Check the **MDV?** checkbox if the main input dataset contains a missing dependent variable column. (See MDV in the Main Mappings panel description for more information.)
- Check the **Steady State** checkbox if the main input dataset contains a steady state column. (See More on Steady State in the Main Mappings panel for more information.)

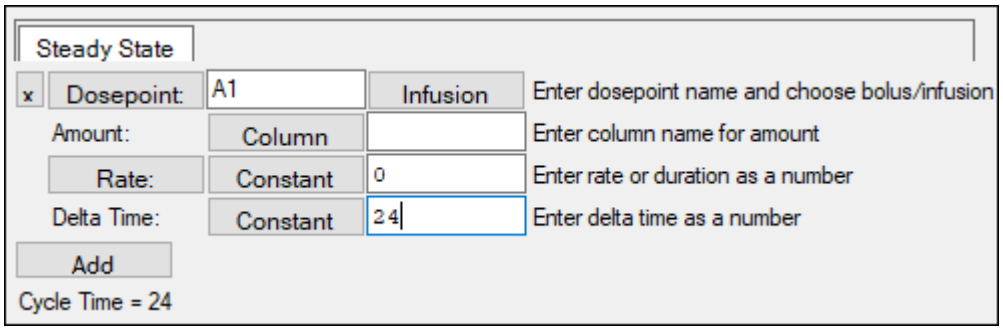
  Selecting **Steady State** displays the Steady State sub tab, which allows users to add new steady state treatment sequence items.



  - The **Dosepoint** button indicates that the main dosepoint will be used on the given compartment. Click this button to use the second dosepoint (the button name changes to **Dosepoint2**), if one is available.
  - In the **Dosepoint** field, type a name for the steady state dose point.
  - Click **Bolus** multiple times to toggle between **Bolus** and **Infusion** as the input type.
  - Click **Constant** (next to **Amount**) multiple times to toggle between **Constant** (the dose amount is a constant value) and **Column** (the dose amount comes from a column in the main input dataset).
  - In the **Amount** field, enter the constant dose amount (when the button displays **Constant**) or the column name (when the button displays **Column**).

- Click **Constant** (next to **Delta Time**) multiple times to toggle between **Constant** (the time between doses is a constant value) and **Column** (the time between doses comes from a column in the main input dataset).

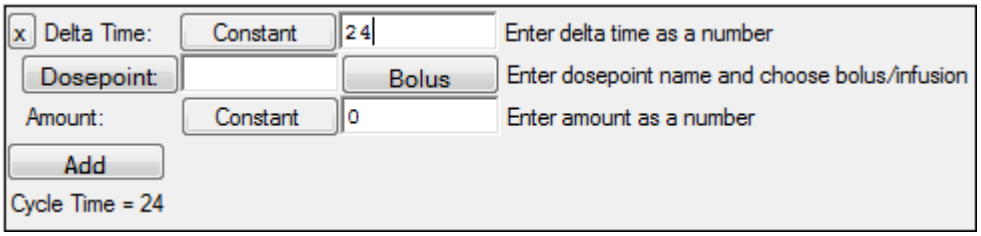
If **Infusion** is selected, an additional **Rate** option is added to the **Steady State** list.



- For the steady-state constant infusion case, the **Amount** should be set to 0. Otherwise, it is the same as what is described above for the case where **Bolus** is selected.
- For the steady-state constant infusion case, enter the infusion rate or column name for the rate in the **Rate** field. Otherwise, click **Rate** multiple times to toggle between **Rate** and **Duration** as the desired input type and, in the field, enter the rate of the infusion or column name for rate (when the button displays **Rate**) or the duration of the infusion or column name for duration (when the button displays **Duration**).
- For the steady-state constant infusion case, the **Delta Time** should be set to 0 (i.e., for this case, non-zero values for the time between doses are not allowed). Otherwise, it is the same as what is described above for the case where **Bolus** is selected.
- Click **Add** to add a new steady state treatment sequence item.
- Click the **X** button to remove a steady state treatment sequence item.

- Check the **ADDL** checkbox if the main input dataset contains an additional identical doses column. (See "More on ADDL" in the Main Mappings panel description for more information.)

  Selecting **ADDL** displays the ADDL sub tab, which allows users to add additional identical doses.



  The options are the same as for the Steady State sub tab. Refer to the Steady State description for instructions on setting these options.

- Check the **Date?** checkbox if the main input dataset contains a date column.
- In the **Date** menu, select the date format used in the dataset.

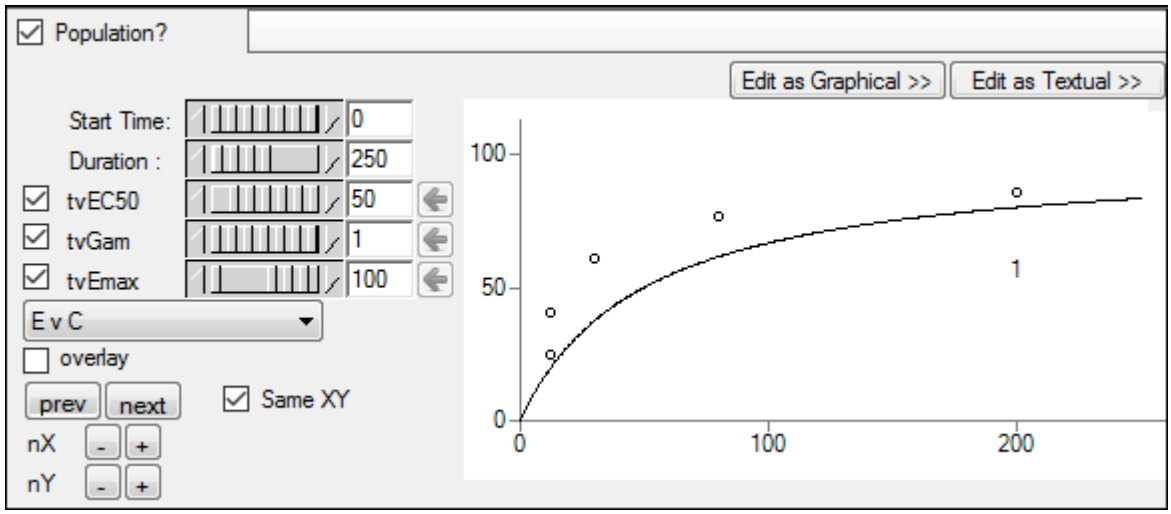
  Date columns in the dataset must be the Text data type. Phoenix automatically converts Dates to the Text data type on import.

- If more than one ID column is mapped (i.e., a set of columns is required to uniquely identify data records from an individual), then the system provides a list of all the ID variables with a bar to their right that allows the user to swap the order of ID pairs.

Note: If Steady State or ADDL input is specified, and if the input column names are given, the column names are case-sensitive and must match the column names in the input dataset.

Initial Estimates tab

The Initial Estimates tab allows users to view plots of different variables in the main input dataset and view parameters in the model. Users can adjust initial estimates for the fixed effects to determine the best values.



- Use the pointer to move the **Start Time** and **Duration** sliders right or left to select which portion of the initial estimates time line to view, or type values into the fields beside the sliders. For example, if the total time line runs from 0–50, but you want to zoom in on 35–45. Set the start time to 35 and the duration to 10.
- Use the pointer to move the fixed effect slider right or left or adjust the initial estimates for each fixed effect, or type the values into the field beside the fixed effect slider.

Note: When typing a decimal value, be aware that typing an initial “0” will cause the checkbox to automatically become unchecked. Either drop the initial “0” when entering the value or recheck the checkbox.

By default, the fixed effect sliders do not allow users to select negative values. Moving the slider to the left changes the initial estimates to smaller decimal values. Clear the checkbox beside each fixed effect to allow the initial estimate to change to negative values.

- If different initial estimates are selected in the Initial Estimates tab, click the **Copy to initial estimates** < button to copy the initial estimate value to the **Initial** field for each parameter in the Fixed Effects tab.

  If the initial estimates in the Fixed Effects tab match the initial estimates in the Initial Estimates tab, the **Copy to initial estimates** button is not available.
- In the plots menu C vs time, select the different variables, parameters, and covariates used to create the plots.

  The plots available in the Initial Estimates tab change depending on the type of dataset used, the

type of model selected, and any additional parameters and covariates that are selected. For example, in a one-compartment PK model with IV bolus dosing and no extra parameters or covariates, users can select plots of C vs time, V vs time, Ke vs time, V vs C, and Ke vs C.

- Check the **log** checkbox if the Y axis is logarithmic.
- Check the **overlay** checkbox to create one plot that contains all profiles (**will** remove the previous, next, and trellis controls).
- Click **prev** to view the previous plot.
- Click **next** to view the next plot.
- Check the **Same XY** checkbox to use the same tick mark values on the X and Y axes.
- Click **nX (+)** to add a trellis row. This displays extra plots in the Initial Estimates tab.
- Click **nX (-)** to remove a trellis row.
- Click **nY (+)** to add a trellis column. This displays extra plots in the Initial Estimates tab.
- Click **nY (-)** to remove a trellis column.
- Increase the size of the Initial Estimates tab to expand the plot view.

Note: If the plotting window is not refreshed after making a change to the model structure or mappings, select a different tab and then go back to the Initial Estimates tab.

Run Options tab

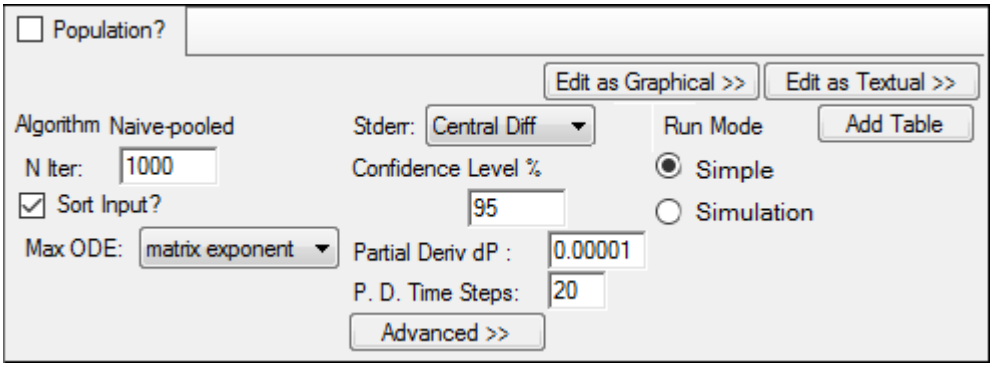
The Run Options tab allows users to determine how Phoenix executes a model. For individual modeling, the jobs are always run locally using the run method **Naive-pooled**, which cannot be changed. Population modelers have many more run options.

Individual modeling run options
Population modeling run options

For detailed explanations of Maximum Likelihood Models run methods, see “Run modes”.

Individual modeling run options

The following options are displayed on the Run Options tab when the **Population?** checkbox is unchecked.



- In the **N Iter** field, type the maximum number of iterations to use with each modeling run (the default is 1000 and the maximum is 10,000).
- Check the **Sort Input?** checkbox (the default) to sort the input by subject and time values.

  When checked, the data is automatically sorted by ID (up to 5 levels of ID) and then by Time (if the model is time-based). Unchecking this option will process the data in the order given in the input dataset, and thus if the same subject ID is present in the dataset but the records are not consecutive they will be treated as two different subjects.

  If a single ID entry within a column is non-numeric then the entire column is considered non-numeric, otherwise it is considered numeric. Note that non-numeric and numeric ID variables are sorted differently. For example, in numeric ID sorting, 2 comes before 10 but in non-numeric order 10 comes before 2.

  If dosing information comes from a dosing worksheet (external or internal) then **Sort Input?** is automatically selected because of the need to merge the dosing and main worksheets. Similarly, if the model has a parameter worksheet that provides initial values, the **Sort Input?** option is automatically selected because it requires merging and thus sorting over sub-populations or individuals.

  The **Sort Input?** option is available for population modeling as well.
- In the **Max ODE** menu, select one of the ODE (ordinary differential equations) solver methods: **matrix exponent**, **auto-detect**, **non-stiff DVERK**, **stiff**, **non-stiff DOPRI5**.

  For more on ODE methodology, see “Differential equations in NLME”.

  Standard errors for the parameter estimators for individual models are computed using the Hes-

sian method. The Hessian method evaluates the uncertainty matrix as R-1, where R-1 is the inverse of the second derivative matrix of the -2*Log Likelihood function.

Note: Computing the standard errors for a model can be done as a separate step after model fitting. This allows the user to review the results of the model fitting before spending the time computing standard errors.
For engines other than QRPEM:
– After doing a model fitting, accept all of the final estimates of the fitting.
– Set the number of iterations to zero.
– Rerun with a method selected for the **Stderr** option.
For QRPEM, use the same steps, but also request around 15 burn-in iterations.

- In the **Stderr** menu, select the method used to compute standard errors.

  none (no standard error calculations are performed)
  Central Diff for the second-order derivative of f uses the form:

$$(f(x+h)-2f(x)+f(x-h))/h^2$$

  Forward Diff for the second-order derivative of f uses the form:

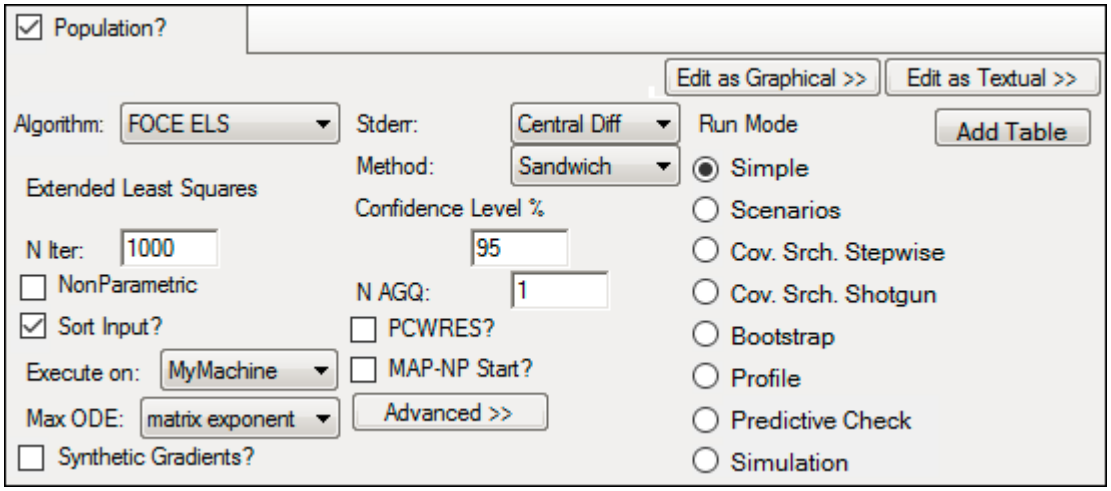
$$(f(x+2h)-2f(x+h)+f(x))/h^2$$

  The **Stderr** option is available for population modeling as well.
- In the **Confidence Level %** field, enter the confidence interval percentage.
- In the **Partial Deriv dP** field, enter the amount of perturbation to apply to parameters to obtain partial derivatives.
- Enter the number of time steps in the **P. D. Time Steps** field.

  Plots of the partial derivatives are spaced at this number of time vales over the whole time line. To get a coarse plot, specify a low number. To get a smoother plot, specify a larger number of steps.

Population modeling run options

The following options are displayed in the Run Options tab when the **Population?** checkbox is checked.



Not all run options are applicable for every run method. Some options are made available or unavailable depending on the selected run method. For detailed explanations of Maximum Likelihood Models run options, see "Run modes".

- In the **Algorithm** menu, select one of seven run methods:

  FOCE L-B (First-Order Conditional Estimation, Lindstrom-Bates)
  FOCE ELS (FOCE Extended Least Squares)
  FO (First Order)
  Laplacian
  Naive pooled
  IT2S-EM (Iterated two-stage expectation-maximization)
  QRPEM (Quasi-Random Parametric Expectation Maximization)

- In the **N Iter** field, type the maximum number of iterations to use with each modeling run (the default is 1000).
- Check the **NonParametric** checkbox to use the NonParametric engine for producing nonparametric results in the output. This option is available for **FOCE L-B**, **FOCE ELS**, **Laplacian**, **IT2S-EM**, and **QRPEM** methods.
  - If the **NonParametric** checkbox is selected, then the **N NonPar** field is made available.
  - In the **N NonPar** field, type the maximum number of iterations of nonparametric computations to complete during the modeling process.
- Check the **Sort Input?** checkbox (the default) to sort the input by subject and time values. (Refer to the **Sort Input?** description in the Individual Modeling section for additional information.)
- Select the local or remote machine or grid on which to execute the job from the **Execute on** menu.

  The contents of this menu can be edited, refer to "Compute Grid preferences".

Note: When a grid is selected, loading the grid can take some time and it may seem that the application has stopped working.

Note: Make sure that you have adequate disk space on the grid for execution of all jobs. A job will fail on the grid if it runs out of disk space.

- In the **Max ODE** menu, select one of the ODE (ordinary differential equations) solver methods. For more on ODE methodology, see "Differential equations in NLME".
- Check the **Synthetic Gradients?** checkbox to allow synthetic gradients in the model.

  When this is selected in population modeling, the engine computes and makes use of analytic gradients with respect to etas. In some cases, with differential equation models, one or more gradient components can be computed by integrating the sensitivity equations along with the original model system using a numerical differential equation solver. For reasonably sized models, selecting **Synthetic Gradients** can help speed and accuracy.
- In the **Stderr** menu, select the form of standard error to compute. (Refer to the **Stderr** description in the Individual Modeling section for additional information.)

  none (no standard error calculations are performed)
  Central Diff, that uses the form:

$$(f(x+h)-2f(x)+f(x-h))/h^2$$

Forward Diff, that uses the form:

$$(f(x+2h)-2f(x+h)+f(x))/(h^2)$$

- Select the **Method** of computing the standard errors from the menu:

  Hessian: The Hessian method of parameter uncertainty estimation evaluates the uncertainty matrix as R-1, where R-1 is the inverse of the second derivative matrix of the -2*Log Likelihood function. This is the only method available for individual models.

  Sandwich: The sandwich method has both advantages and disadvantages relative to the Hessian method. The main advantage is that, in simple cases, the sandwich method is robust for covariance model misspecification, but not for mean model misspecification. The main disadvantage is that it can be less efficient than the simpler Hessian-based method when the model is correctly specified.

  Fisher Score: The Fisher Score method is fast and robust, but less precise than the Sandwich and Hessian methods.

  Auto-detect: When selected, NLME automatically chooses the standard error calculation method. Specifically, if both Hessian and Fisher score methods are successful, then it uses the Sandwich method. Otherwise, it uses either the Hessian method or the Fisher score method, depending on which method is successful. The user can check the Core Status text output to see which method is used.

- In the **Confidence Level %** field, enter the confidence interval percentage or use the default.
- In the **N AGQ** field, enter the maximum number of integration points for the Adaptive Gaussian Quadrature. This field is available for **FOCE ELS** and **Laplacian** methods.
- In the **ISAMPLE** field, enter the number of sample points or use the default. For each subject, the joint likelihood, the mean, and the covariance integrals will all be evaluated with this number of samples from the designated importance sampling distribution. Increasing this number improves the accuracy of the integrals. This option is available for the **QRPEM** method.
- Check the **PCWRES?** checkbox to generate the predictive check weighted residuals in the results. This option is available for **FOCE L-B**, **FOCE ELS**, **FO**, **Laplacian**, and **IT2S-EM** methods. It is unchecked by default. When checked, then the **nrep** field becomes available. Enter the number of replicates or use the default for predictive check weighted results. The maximum number of replicates is 1001, a number higher than this is automatically reduced, by the system, to 1001 and a warning appears in the Core Status tab.

  PCWRES is an adaptation of France Mentre' et al.'s normalized predictive error distribution method. Each observation for each individual is converted into a nominally normal N(0,1) random value. It is similar to WRES and CWRES, but computed with a different covariance matrix. The covariance matrix for PCWRES is computed from simulations using the time points in the data.
- Check the **MAP-NP Start?** checkbox to perform Maximum A Posteriori initial Naive Pooling. This option is available for **FOCE L-B**, **FOCE ELS**, **FO**, **Laplacian**, **QRPEM** and **IT2S-EM** methods. If selected, an **Iterations** field appears requesting the number of iterations to apply MAP-NP.

  This process is designed to improve the initial fixed effect solution with relatively minor computational effort before a main engine starts. It consists of alternating between a Naive pooled engine computation, where the random effects (etas) are fixed to a previously computed set of values, and a MAP computation to calculate the optimal MAP eta values for the given fixed effect values that were identified during the Naive Pooled step.

  On the first iteration, MAP-NP applies the standard Naive pooled engine with all etas frozen to zero to find the maximum likelihood estimates of all the fixed effects. The fixed effects are then

frozen and a MAP computation of the modes of the posterior distribution for the current fixed effects parameters is then performed. This cycle of a Naive pooled followed by MAP computation is performed for the number of requested iterations. Note that, throughout the MAP-NP iteration sequence, the user-specified initial values of an eps and Omega are kept frozen.

- Check the **FOCEHess?** checkbox to use FOCE Hessian. This option is available for **Laplacian** and **IT2S-EM** methods.
- Click **Advanced** to toggle access to advanced run options for population models (only).

  All engines have the following Advanced options.
  - In the **LAGL nDig** field, enter the number of significant decimal digits for the LAGL algorithm to use to reach convergence. Used with FOCE ELS and Laplacian Run methods. LAGL, or LaPlacian General Likelihood, is a top level log likelihood optimization that applies to a log likelihood approximation summed over all subjects.
  - In the **SE Step** field, enter the standard error numerical differentiation step size. SE Step is the relative step size to use for computing numerical second derivatives of the overall log likelihood function for model parameters when computing standard errors. This value affects all Run methods except IT2S-EM, which does not compute standard errors.
  - In the **BLUP nDig** field, enter the number of significant decimal digits for the BLUP estimation to use to reach convergence. Used with all run methods except **Naive pooled**. BLUP, or Best Linear Unbiased Predictor, is an inner optimization that is done for a local log likelihood for each subject. BLUP optimization is done many times over during a modeling run.
  - In the **Modlinz Step** field, enter the model linearization numerical differentiation step size. **Modlinz Step** is the step size used for numerical differentiation when linearizing the model function in the FOCE approximation. This option is used by the FOCE ELS and FOCE L-B Run methods, the IT2S-EM method when applied to models with Gaussian observations, and the Laplacian method when the FOCEhess option is selected and the model has Gaussian observations.
  - In the **ODE Rel. Tol.** field, enter the relative tolerance value for the **Max ODE**.
  - In the **ODE Abs. Tol.** field, enter the absolute tolerance value for the **Max ODE**.
  - In the **ODE max step** field, enter the maximum number of steps for the **Max ODE**.

The following are additional advanced options available only for the **QRPEM** method.

- Check the **MAP Assist** checkbox to perform a maximization of the joint log likelihood before each evaluation of the underlying integrals in order to find the mode. The importance sampling distribution is centered at this mode value. (If **MAP Assist** is not selected, the mode finding optimization is only done on the first iteration and subsequent iterations use the mean of the conditional distribution as found on the previous iteration.) If the **MAP Assist** checkbox is checked, enter the periodicity for MAP assist in the **period** field. (For example, a value of two specifies that MAP assist should be used every other iteration.)
- Select the importance sampling distribution type from the **Imp Samp Type** pull-down menu.

  normal: Multivariate normal (MVN)

  double-exponential: Multivariate Laplace (MVL). The decay rate is exponential in the negative of the sum of absolute values of the sample components. The distribution is not spherically symmetric, but concentrated along the axes defined by the eigenvectors of the covariance matrix. MVL is much faster to compute than MVT.

  direct: Direct sampling.

  T: Multivariate t (MVT). The MVT decay rate is governed by the degrees of freedom: lower values correspond to slower decay and fatter tails. Enter the number of degrees of freedom in the **Imp**

Samp DOF field. A value between four and 10 is recommended, although any value between three and 30 is valid.

mixture-2: Two-component defensive mixture. (See T. Hesterberg, "Weighted average importance sampling and defensive mixture distributions," Tech. report no. 148, Division of Biostatistics, Stanford University, 1991). Both components are Gaussian, have equal mixture weights of 0.5, and are centered at the previous iteration estimate of the posterior mean. both components have a variance covariance matrix, which is a scaled version of the estimated posterior variance covariance matrix from the previous iteration. One component uses a scale factor of 1.0, while the other uses a scale factor determined by the acceptance ratio.

mixture-3: Three-component defensive mixture. Similar to the two-component case, but with equal mixture weights of 1/3 and scale factors of 1, 2, and the factor determined by the acceptance ratio.

- Check the **MCPEM** checkbox to use Monte-Carlo sampling instead of Quasi-Random. Although the default is recommended, using Monte-Carlo sampling may be necessary if the most direct comparison possible with other MCPEM algorithms is desired.
- Check the **Run all** checkbox to execute all requested iterations, ignoring the convergence criteria.
- In the **# SIR Samp** field, enter the number of samples per subject used in the Sampling Importance Re-Sampling algorithm to determine the number of SIR samples taken from the empirical discrete distribution that approximates the target conditional distribution.

  The **# SIR Samp** from each subject are merged to form the basis for a log-likelihood optimization that allows fixed effects that are not paired with random effects to be estimated. The **# SIR Samp** is usually far smaller than the number that would be used if the SIR algorithm were not used.

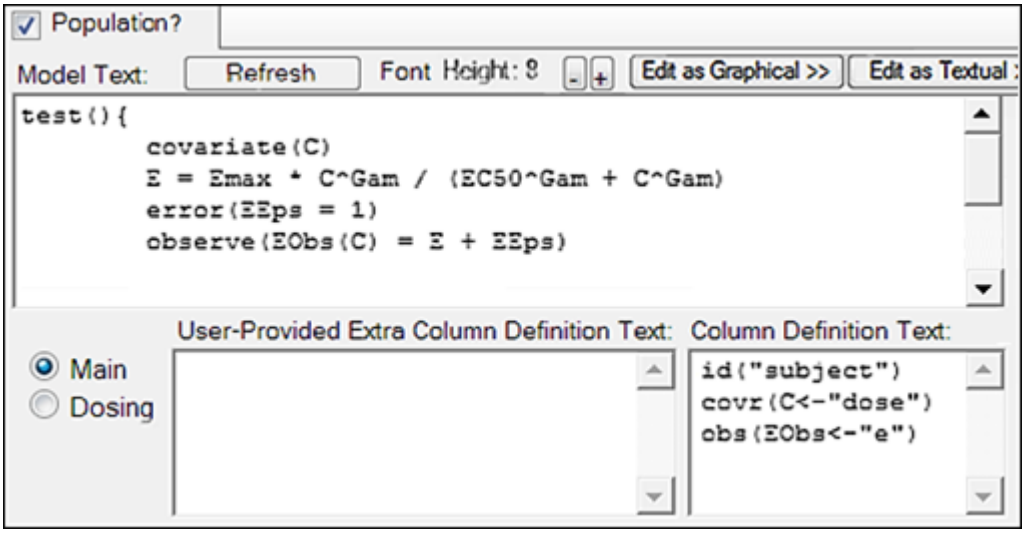
  The default of 10 is usually adequate, unless the number of subjects is extremely small. If necessary, **# SIR Samp** can be increased up to a maximum of **ISAMPLE**.
- In the **# burn-in Iters** field, type the number of QRPEM burn-in iterations to perform at startup to adjust certain internal parameters. During a burn-in iteration, importance sampling is performed and the three main integrals are evaluated, but no changes are made to the parameters to be estimated or the likelihood. The default is zero, in which case the QRPEM algorithm starts normally. Typical values, other than zero, are 10 to 15.
- Check the **Frozen omega burn-in** checkbox to freeze the omega but not theta for the number of iterations entered in the **# burn-in Iters** field. If this checkbox is not checked, then both omega and theta are frozen during the burn-in.
- Check the **Use previous posteriors** checkbox to start the model up from the posteriors and eta-means saved from a previous run. Checking this box allows the QRPEM engine to be restarted from a previous run using the final posteriors from that run as initial posteriors in the restart, thus allowing the run to resume from exactly where it left off. To do this, press **Accept All Fixed+Random** in the Parameters tab to accept the final parameter estimates from the first run as initial parameter estimates for the new run, then run QRPEM with the **Use previous posteriors** box checked.
- In the **Acceptance ratio** field, enter a decimal value for the acceptance ratio (default is 0.1). As this value is reduced below 0.1, the importance sampling distribution becomes broader relative to the conditional density.

Note: **ISAMPLE**, **Imp Samp Type**, and **Acceptance ratio** can all be used to increase or decrease the coverage of the tails of the target conditional distribution by the importance sampling distribution.

- Select the Quasi-Random scrambling method to use from the **Scramble** menu: **none**, **Owen**, or **TF** (Tezuka-Faur).

Model Text tab

The Model Text tab displays the PML (Phoenix Modeling Language) code that is generated when a model is specified. Phoenix generates the PML code based on the options selected in the Structural and Parameters tabs. The Model Text tab also automatically provides column definitions for the model.



The PML code is displayed in the Model Text field. The PML code updates automatically when any changes are made in the model.

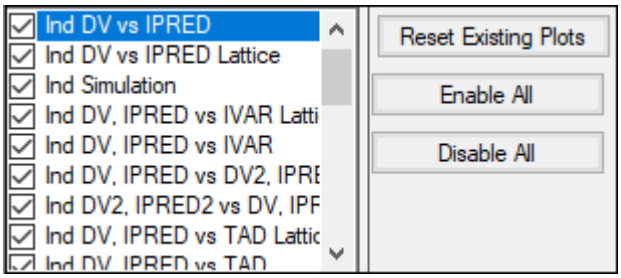
- Click **Refresh** to update the PML code.
- Click **(-)** to decrease the size of the model text and the column definition text.
- Click **(+)** to increase the size of the model text and the column definition text.

  PML code in the Model Text field cannot be edited. The PML code can only be edited when the Text Model mode is selected. The column mappings are displayed in the Column Definition Text field and cannot be edited. They are created from the mappings in the Main Mappings panel and the Dosing panel.
- Select the **Main** option button to view the column definition text for the main mapping contexts.
- Select the **Dosing** option button to view the column definition text for the dose mapping contexts.
- In the **User-Provided Extra Column Definition Text** field, type any extra column mappings that are not automatically generated in the Column Definition Text field.

  Users are not expected to add extra column definitions in the Model Text tab unless it becomes necessary, for example, to use a column for two separate contexts in a model.

Plots tab

The Plots tab allows users to select individual plots to include in the output.



- Use the checkboxes to toggle the creation of graphs.
- Click **Reset Existing Plots** to clear all existing plot output.

  Each plot in the Results tab is a single plot object. Every time a model is executed, each object remains the same, but the values used to create the plot are updated. This way, any styles that are applied to the plots are retained no matter how many times the model is executed.

  Clicking **Reset Existing Plots** removes the plot objects from the Results tab, which clears any custom changes made to the plot display.
- Use the **Enable All** and **Disable All** buttons to check or clear all checkboxes for all plots in the list.

WARNINGS/no warnings tab

The WARNNGS/no warnings tab displays any problems with the model data mappings and settings. If there are any problems, the tab is labeled **WARNINGS**. If there are no problems, the tab is labeled **no warnings**.

Graphical model interface

Note: The graphical model interface is a free-form model-building interface. It is much more sophisticated than the built-in model, which means user error is more likely. It is recommended for advanced users.

To use the graphical editor to create the structural model:

- Click **Edit as Graphical**.
- In the displayed dialog, click **Yes** to confirm using the graphical editor.
- If **Closed form?** is currently checked, a second dialog is presented notifying the user that closed-form will not be used. This is because graphical models are built with differential equations. Click **Yes** to continue.

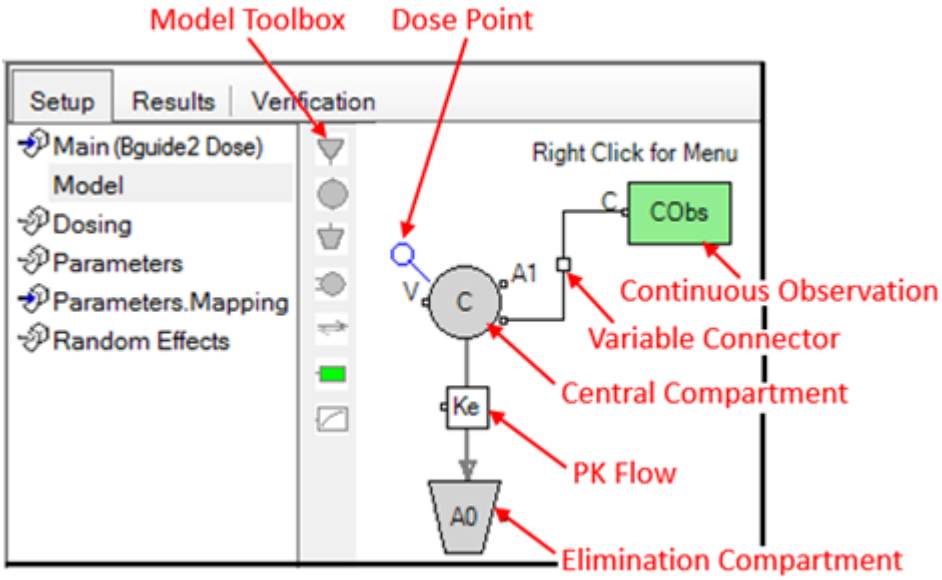


Figure 9-1. Example of a one-compartment, 1st order absorption PK model

When the graphical editor is in use, the **Edit as Graphical** button changes to **Use Builtin**. Click this button to return to the built-in model interface.

The Main Mappings, Dosing, Parameters, and Parameters.Mappings panels and all tabs except the Structure tab work the same in the Graphical model as they do in the built-in model. If no model elements are selected in the Model panel then the Structure tab displays the model parameters, statements, and scale control. Use the pointer to move the slider right or left to adjust the graphical model scale, which makes the model picture larger or smaller. If a compartment is selected, then the compartment type and type-specific settings are displayed in the Structure tab.

Users can specify the model using the Model toolbox or the right-click menu. The right-click **Insert** menu allows users to insert compartments, flows, observations, PD blocks, structural parameters, and other model elements. Users can also cut, copy, paste, and manipulate model elements. (The difference between **Replace** and **Insert** is that the former replaces the selected objects in the diagram with the contents of the clipboard while trying to retain wire and flow connections. The latter just copies the contents of the clipboard into the diagram.)

Insertable model elements include:

Absorption compartment ( ) (See Absorption compartment options.)

Central compartment () (See Central compartment options.)

Peripheral compartment () (See Peripheral compartment options.)

Elimination compartment () (See Elimination compartment options.)

Flow () (See PK Flow.)

Continuous observations () (See Continuous observation block.)

Categorical observations (See Categorical observation block.)

Event observations (See Event observation block.)

Count observations (See Count observation block.)

LL observations (See Log-likelihood observation block.)

Emax () (See Emax block.)

Linear (See Linear block.)

Indirect (See Indirect block.)

Effect Cpt (See Effect compartment block.)

Parameter (See Parameter block.)

Procedure (See Procedure block.)

Expression (See Expression block.)

Annotation (See Annotation block.)

PBPK (See Vascular flow.)

Caution: When switching from a PK/Emax, or PK/Indirect built-in to a Graphical model, verify the tab settings. In particular, the **Freeze PK?** checkbox, when checked in the built-in model, can become unchecked when switching to Graphical.

Absorption compartment options

This PK compartment is used as an input site for doses following a first-order input function. By default, flows connecting an absorption compartment to another compartment use a first-order function. That function can be changed by specifying PK flow options in the Structure tab. Connect an absorption compartment to another compartment via a PK flow to create a 1st-order absorption PK model.

- Use the Model toolbox or select **Insert > Compartment > Absorption** from the right-click menu.
- Absorption compartment work the same as the central compartment, (except there is no **Conc** option to add volume and concentration parameters). Refer to "Central compartment options" for details on setting these parameters.

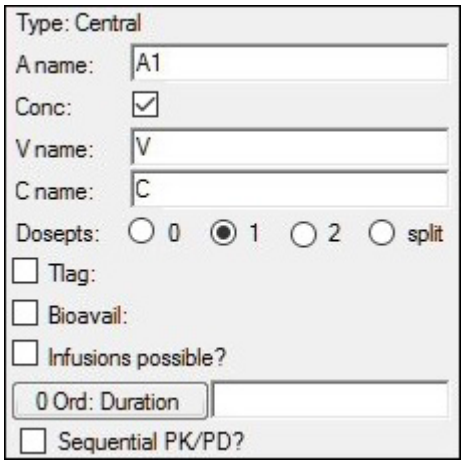
Caution: Be careful not to use the same names for the volume and concentration parameters in different compartments.

- Drag the dosing variable square from the absorption compartment to the central compartment dosing variable square.

Central compartment options

Central compartments can be connected to other PK compartments, including at least one elimination compartment (unless no drug leaves the body), by using a PK flow. Any number of central and peripheral compartments can be connected together, with one exception: if a set of PK compartments using volume parameters are connected by PK flows parameterized in microconstants, there can be only one central compartment.

- Use the Model toolbox or select **Insert > Compartment > Central** from the right-click menu.



- If desired, modify the default dosing variable name in the **A name** field.

  Typically, **A1** is used for intravenous input and **Aa** is used for extravascular input.

- Check the **Conc** checkbox (the default) if the compartment has concentration or volume parameters; uncheck it if there are no concentration or volume parameters.

  The **Conc** checkbox controls the parameterization of the compartment. Selecting it adds the volume parameter **V** and the concentration parameter **C** to the model. When the checkbox is cleared, the compartment has no volume or concentration parameters, and can be useful for certain compartmental PD models.

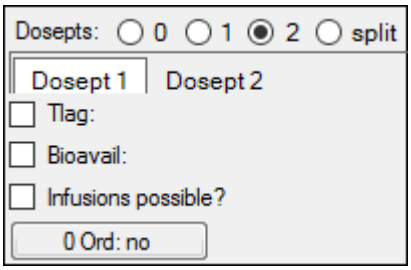
- In the fields for the other structural parameters, enter names for each parameter.

- Select a **Dosepts** option button (**0**, **1**, **2**, or **split**) to specify the number of dose points used by the central compartment. Select **split** to split dosing into the central compartment between two dosepoints.

  Dose points can be defined for any PK compartment. The time(s) and amount(s) of doses are defined in the dataset. Each drug can have multiple formulations. For example, a drug can have one formulation dosing into the central compartment (IV) and another formulation dosing into an absorption compartment (oral or intramuscular). A drug model can include any number of different formulations and different drugs.

Note: Option **2** requires the dosing data to be in separate columns in the input dataset. Option **split** requires the data to be in the same column.

If **1**, **2**, or **split** dose points are selected, users can add time delay parameters, bioavailability expressions, add infusions, and specify zero-order absorption options. For **2** or **split**, these parameters can be set for each dose point by selecting the Dosept 1 and Dosept 2 tabs.



- Check the **Tlag** checkbox to add a time delay parameter to the model. Enter the time lag expression in the field.
- Check the **Bioavail** checkbox to measure bioavailability in the central compartment. Enter the bioavailability expression in the field.

  Bioavailability expressions can be entered as a numerical value or a parameter. For example, if a user enters 0.5, half the drug goes into the compartment.

  To estimate bioavailability, a parameter must be used instead of a number. For example, a user can type F (fraction of dose absorbed parameter) in the Bioavail field to estimate drug availability. If new parameter names are used in the expression, these will have to be inserted into the model. See "Parameter block" for Parameter block usage instructions.

  If users want to estimate bioavailability using multiple dosing routes, then a second parameter like 1-F can be entered in the Dosept 2 tab. Using multiple dosing routes and bioavailability parameters can be useful when modeling first- and second-pass absorption rates.
- Check the **Infusions possible?** checkbox if the dosing uses infusions.
  - A dosing parameter rate, such as **Aa Rate** or **A1 Rate**, is added to the contexts in the Main Mappings panel.
  - A **Duration?** checkbox is also added in the Structure tab. Checking this box causes the context **A1** or **Aa Rate** to change to **A1** or **Aa Duration**.
- Click **0 Ord: No** multiple times to toggle through the zero-order options:

  No: There is no zero-order input.
  Rate: The drug is introduced into the system at a constant rate. Enter the zero-order rate expression in the field, as either a parameter or a numeric value.
  Duration: The drug is introduced over a finite period of time, or duration. Enter the zero-order duration expression in the field, as either a parameter or a numeric value.
- Select the **Sequential PK/PD?** checkbox if the PK model is part of a PK/PD model that is being fitted sequentially.This will freeze the PK portion of the model and turn its random effects into covariates. See "Sequential PK/PD population model fitting" for more information.

Peripheral compartment options

Peripheral compartments can be connected to other PK compartments, including central and elimination compartments, by using a PK flow. If there is only one central compartment, the volumes of the peripheral compartments are determined by their surrounding constraints, or flow parameters.

- Use the Model toolbox or select **Insert > Compartment > Peripheral** from the right-click menu.
- Peripheral compartment options work the same as the central and absorption compartment options. Refer to “Central compartment options” for details on setting these parameters.

Elimination compartment options

The elimination compartment (which is shown as a `urinecpt` code statement) can be connected to any other PK compartment via a PK flow. The only difference between `urinecpt` and `deriv` is that `urinecpt` is ignored when determining a steady-state condition.

- Use the Model toolbox or select **Insert > Compartment > Elimination** from the right-click menu.
- Elimination compartment options work the same as the central compartment options, (except there is no **Conc** option to add volume and concentration parameters). Refer to “Central compartment options” for details on setting these parameters.
- The fraction excreted (**Fe**) is an additional parameter that can be include in the code statement, as shown in the following example statement:

```
urinecpt(A0=(A1*Ke)
  ,fe=Fe
)
```

- To indicate that the elimination compartment is set to zero right after being observed, edit the model as textual and change the PML code to:

```
observe(UrineObs=A0+eps, doafter={A0=0; })
```

PK Flow

Use a PK flow to represent a mass flow between PK compartments. To connect two PK compartments with a flow:

- Click the flow button in the Model toolbox, or right-click and select **Insert > Flow**.
- Use the pointer to click on the first and second PK compartments of the flow.

  Note that lag time for an absorption model is indicated in the absorption or central compartments.

Different types of default PK flows are added depending on which compartments they connect:

- If the absorption compartment is connected to a central, peripheral, or elimination compartment, the default PK flow is a micro parameter, one-way flow. The default parameter name for the PK flow changes depending on the direction on which the flow moves, and on which compartments are being connected. For example, if the flow moves from the absorption compartment to another compartment, the parameter name for the forward movement (Kfwd) is Ka. The default parameter name for any backward PK flow is K12.
- The default PK flow between the central or peripheral compartments and the elimination compartment is named CL, and it is a clearance/volume parameter, one-way flow.
- The default PK flow between the central and peripheral compartments is named Q, and it is a clearance/volume parameter, with a two-way flow.

To specify PK flow options

- Use the pointer to select a PK flow.
- Click **Parm** multiple times to toggle through the parameterization options for the model.

  Micro: Enter a variable name for the flow into the second compartment in the **Kfwd** field or use the default. If the **2Way** checkbox is also checked, enter a variable name for the flow back into the first compartment in the **Kbak** field.

  Clearance/Volume: Enter a variable name for the clearance in the **CL** field or use the default.

  Saturating: Enter variable names in the **Vmax** field for the maximum metabolic rate and in the **Km** field for the fractional metabolic rate or use the defaults.
- Click the **2Way** checkbox to change the PK flow to a two-way flow; uncheck it to create a one-way flow.
- Click **Draw** multiple times to toggle through the options for drawing the PK flow in the Graphical model interface.

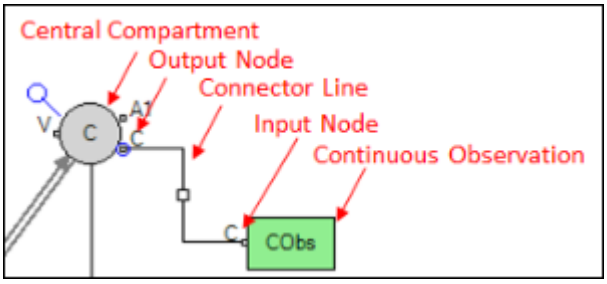
  Diagonal: Draws the diagonal flow.
  Horizontal First: Draws the flow horizontally from the first compartment.
  Vertical First: Draws the flow vertically from the first compartment.

Continuous observation block

The Continuous Observation block links a model variable such as excreted amount, or plasma concentration, to observed data. The observations are normally distributed around the value of the model variable with some standard deviation prescribed by the error model. When estimating model parameters, the likelihood of an observation is obtained from the normal distribution. When simulating data, the observations are generated by sampling values from the standard normal distribution and applying them to the terms in the error model.

- Insert a Continuous Observation block by using the Model toolbox or selecting **Insert > Observation > Continuous** from the right-click menu.
- Place the pointer over an output node on a compartment or a block. The connector is surrounded by a blue circle when it is selected. The Continuous Observation block can be connected to any PK compartment, observation block, PD block, or parameter block.
- Drag the output node connector to the C input node on the Continuous Observation block.

  A line is displayed that represents the connection between the Continuous Observation block and the compartment or block.



To delete a connection

- Select the square in the middle of the connector line. The line is surrounded by a blue square when it is highlighted.

- Right-click the square and select **Insert > Delete**.

To specify continuous observation options

- Use the pointer to select a Continuous Observation block.
- In the **Name** field, enter a name for the Continuous Observation block or use the default name of CObs.
- The rest of the options are the same as in the built-In model.

Categorical observation block

Use the Categorical Observation block to model multinomial data. The data should be given as integer (whole) number values, such as 0, 1, 2, etc. A key feature of the Categorical Observation block is the ability to use model variables to affect the probability of observing data in a particular category. For instance, the Categorical Observation block could be used to link the probability of a patient reporting adverse effects of a given type to the administered dose or even drug concentration.

The default settings for the Categorical Observation block give a 50% chance of a zero or one output when the input value is zero. The probability of a one output increases as the input value increases from zero.

- Insert a Categorical Observation block by selecting **Insert > Observation > Categorical**.

The Categorical Observation block can be connected to any PK compartment, observation block, PD block, or parameter block. See the Continuous Observation block section for instructions on adding and deleting a connection.

To specify categorical observation options

- Use the pointer to select a Categorical Observation block.
- In the **Mtn** field, enter a name for the Categorical Observation block or use the default name.
- Check the **Inhibitory** checkbox to set the output values to decrease as the input increases.
- In the **Link** menu, select the inverse link function.

  This function determines the shape of the demarcation between output values as a function of inputs. This is the inverse of the link function sometimes used in fitting data. In the equations listed below, ***i*** is the input variable value, ***o*** is the offset between a given set of outputs such as zero or one, and ***p*** is the probability of getting a specific output, given ***i*** and ***o***.

  Inverse link functions include:

  Logit: Inverse of the sigmoid function. $p=e^{i+o}/(1+e^{i+o})$
  Probit: Inverse cumulative distribution function. $p=G(i+o)$ where G is the cumulative normal distribution.
  Log-log: Logarithmic function. $p=\exp(-e^{-i+o})$
  CLog-log: Complementary logarithmic function. $p=i+o$ (truncated to the interval [0,1])
- In the **Slope** field, type the slope expression.
- In the **N outcomes** menu, click (**-**) to decrease the number of outcomes, and click (**+**) to increase the number of outcomes.

  N outcomes corresponds to the number of categorical output values. For example, if N outcomes

is three, the output could equal zero, one, or two, depending on the input value. The minimum number of outcomes is two.

- In the **Icept** field, type the intercept expression.

  Enter an expression representing the negative value of the input value that gives a 50% probability of outputting either of the adjacent output values. This is the negative of the offset between outputs. Because it is the negative, the values should be entered in descending numerical order. Use the syntax for expressions, described under "Expression block".

  Each extra outcome that is added to the categorical observation adds an extra Icept field. The first Icept field is labeled Icept10. The second Icept field is labeled Icept21. Each extra Icept field is labeled Icept32, Icept43, Icept54, and so forth.

  For more information on the use of the Categorical Observation block, see "Multi statement for Categorical models".

  Make sure the Model Text tab shows an entry for the observation in the **Column Definition Text** field. It is displayed as ***"obs([block name]<-["column name"])***. If there is no entry, type it in the **User-Provided Extra Column Definition Text** field.

Event observation block

An Event block represents an unscheduled event. Observations take the form of indicator variables for the event: 1 if the event occurs at the time of the observation, or 0 (zero) if the event has not yet occurred at the time of the observation.

The likelihood of the observation is determined by integrating the hazard over the length of the sampling interval, so that at the end of the interval the expected number of events are computed. The expected, or mean, number of events is made the mean of a Poisson distribution. A sample is taken at random from that Poisson distribution at each sampling interval to say how many, if any, events occurred.

The number of events during each sampling interval, which is only one if a probability is entered, is added to the N output variable of the event block.

An event block represents an unscheduled event. One record is created in the results for each evaluation period during which the event is triggered. Each record indicates the time, the censor value (one for event occurrence; zero for right-censored), and a sequence number showing the total number of events having occurred for a given subject.

Note: Censor values in the results are zero for censored events and one for non-censored event records, where the event happened.

- Insert an Event Observation block by selecting **Insert > Observation > Event** from the right-click menu.

The Event Observation block can be connected to any PK compartment, observation block, PD block, or parameter block. See the Continuous Observation block section for instructions on adding and deleting a connection.

To specify event observation block options

- Use the pointer to select an Event Observation block.
- In the **Evt** field, enter a name for the Event Observation block or use the default.

- Check the **Has Input?** checkbox if the slope term comes from elsewhere in the model and enter the hazard slope expression in the **Slope** field.

  An input node labeled C is added to the Event Observation block.

- In the **Icept** field, type the base hazard expression.

  Enter an expression representing the hazard rate for the event, that is, the average frequency of event occurrences expected. This is the same as 1.0/(mean time between events). Use the syntax for expressions. The expression must have the units 1/time. For example, a hazard rate of 1/20{h} indicates that one twentieth of an event happens per hour on average (one event in 20 hours).

For more information on the use of the Event Observation block, see “Event statement for Time-to-event models”.

Make sure the Model Text tab shows an entry for the observation in the **Column Definition Text** field. It is displayed as ***obs([block name]<-[“column name”])***. If there is no entry, type it in the **User-Provided Extra Column Definition Text** field.

Count observation block

- Insert a Count Observation block by selecting **Insert > Observation > Count** from the right-click menu.

With the exception of the **Cnt** field versus the **Evt** field, the options are the same as for the Event observation block.

For more information on the use of the Count Observation block, see “Count statement for Count models”.

Log-likelihood observation block

- Insert a Log-likelihood Observation block by selecting **Insert > Observation > LL** from the right-click menu.

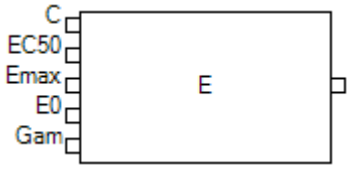
With the exception of the **LL** field versus the **Evt** field, the options are the same as for the Event observation block.

For more information on the use of the Log-likelihood observation block, see “LL statement for user-defined log-likelihood models”.

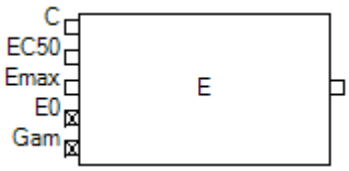
Emax block

The Emax block represents an Emax or a sigmoidal Emax model, where the output is the effect level as a function of drug concentration. The baseline effect level (E0) is always zero.

The Emax block contains five input nodes and one output node.



The input nodes are made available depending on the options that are selected in the Structure tab. If an input node cannot be used, it is marked with an X.



The input nodes on the Emax block can be connected to output nodes on PK compartments, observation blocks, PD blocks, and parameter blocks.

Once an output node is connected its name cannot be changed.

- Insert an Emax block by selecting **Insert > PD > Emax** from the right-click menu.

To specify Emax block options

- Use the pointer to select an Emax block.
- In the **Emx** field, enter a name for the Emax block or use the default.
- Check the **Baseline** checkbox if the model has a baseline response.

  Selecting Baseline adds a new parameter, "first", which has the default value E0 and is the baseline response. The **Fractional** checkbox is also made available.
- Check the **Fractional** checkbox if the Emax model is fractional.

  Selecting **Fractional** modifies the E0 equation statement.
- Check the **Inhibitory** checkbox if the Emax model is inhibitory.

  The structural parameters EC50 and Emax change to IC50 (concentration producing 50% of maximal inhibition) and E0 (baseline effect).
- Check the **Sigmoid** checkbox if the Emax model is sigmoidal.

  A new parameter, Gam, which is a shape parameter, is added.
- In each parameter field, accept the default names or type new names for each parameter.

  Each parameter corresponds to an input node on the Emax block. Possible Emax parameters include:

  C: Drug concentration in plasma, or another input variable.
  EC50: Input level that achieves 50% of predicted maximum effect in an Emax model.
  IC50: Input level required to produce 50% of the maximal inhibition.
  first (E0): Baseline effect.
  EMax: Maximum drug effect.
  IMax: Maximum drug inhibition (requires that both **Baseline** and **Inhibitory** checkboxes be checked).
  Gam: Shape parameter.

Linear block

The Linear block provides a simple linear model in the form: ***Output*=**A****Input*+**B.

The Linear block contains four input nodes and one output node. The input nodes are made available depending on the options that are selected in the Structural tab. If an input node cannot be used, it is marked with an X.

The input nodes on the Linear block can be connected to output nodes on PK compartments, observation blocks, PD blocks, and parameter blocks. Once an output node is connected, its name cannot be changed.

- Insert a linear block by selecting **Insert > PD > Linear** from the right-click menu.

To specify Linear block options

- Use the pointer to select a Linear block.
- In the **order** menu, select the order of the linear model to determine how many parameters the Linear block uses.
- In the **Linear** field, enter a name for the Linear block or use the default.
- In each parameter field, accept the default names or type new names for each parameter.

  Each parameter corresponds to an input node on the Linear block. Possible Linear parameters include:

  C: drug concentration in plasma, or another input variable.
  Alpha: coefficient for zero-order term.
  Beta: coefficient for first-order term (if order >=1).
  Gam: coefficient for second-order term (if order=2).

Indirect block

PK/PD models with stimulation or inhibition of the indirect response formation or degradation can be created using this block.

- Insert an Indirect block by selecting **Insert > PD > Indirect** from the right-click menu.

  The Indirect block contains five input nodes and one output node. The input nodes are made available depending on the options that are selected in the Structural tab. If an input node cannot be used, it is marked with an X.

  The input nodes on the Indirect block can be connected to output nodes on PK compartments, observation blocks, PD blocks, and parameter blocks. Once an output node is connected, its name cannot be changed.
- Insert an Indirect block by selecting **Insert > PD > Indirect** from the right-click menu.

To specify Indirect block options

- Use the pointer to select an Indirect block.
- In the **Ind** field, enter a name for the Indirect block or use the default.
- In the **Indirect** menu select the indirect model type:

  Stim. Limited (limited stimulation of input)
  Stim. Infinite (infinite stimulation of input)
  Inhib. Limited (limited inhibition of input)
  Inhib. Inverse (inverse inhibition of input)
  Stim. Linear (linear stimulation of input)
  Stim. Log Linear (logarithmic and linear stimulation of input)

Use the Indirect menu options to select a model in which the response formation (build-up) or degradation (loss) is stimulated or inhibited by increased concentrations. The default response setting is the build-up of the response, or the production of the response.

- Click **Build-up** multiple times to toggle between the options:

  Build-up: Changes the statement for the Ke parameter to Kin.
  Loss: Changes the statement for the Ke parameter to include Kin-Kout, where Kin is the zero-order input rate constant and Kout is the first output rate constant.

- Click **no Exponent** multiple times to toggle between the options:

  no Exponent: Removes the exponent from the effect statement.
  Exponent: Adds the exponent Gamma to the effect statement.

- In each parameter field, enter names for each parameter or use the defaults.

  Each parameter corresponds to an input node on the Indirect block. Possible Indirect parameters include:

  Kin: Zero-order turnover rate for the production of a response.
  Kout: Fractional turnover rate.
  Emax: Maximum drug induced effect.
  Imax: Maximum drug induced inhibition.
  EC50: Concentration at 50% of maximal effect.
  IC50: Concentration producing 50% of maximal inhibition.
  Gam: Exponent parameter.

Effect compartment block

This block represents an effect site for a PK/PD model with a delay between the central and effect-site concentrations, that is, plots of plasma concentration versus effect level show a counter-clockwise hysteresis. See Sheiner, Stanski, Vozeh, Miller and Ham (1979). Simultaneous modeling of pharmacokinetics and pharmacodynamics: application to d-tubocurarine. *Clin Pharm Ther* 25:358–71 for more about modeling data with a hysteresis in the concentration-effect plot.

The Effect Compartment block contains two input nodes and one output node. The input nodes are made available depending on the options that are selected in the Structural tab. If an input node cannot be used, it is marked with an X.

The input nodes on the Effect Compartment block can be connected to output nodes on PK compartments, observation blocks, PD blocks, and parameter blocks.

Once an output node is connected, its name cannot be changed.

- Insert an Effect Compartment block by selecting **Insert > PD > Effect Cpt** from the right-click menu.

To specify Effect compartment block options

- Use the pointer to select an Effect Compartment block.
- In the **Ce** field, enter a name for the Effect Compartment block or use the default.
- In the **Ke0** field, enter a name or use the default of **Ke0**.

  Ke0 is the exit rate constant from the effect compartment.

Parameter block

The Parameter block is used to add extra structural parameters to a model.

The Parameter block's input nodes are labeled S1, S2, and so forth depending on how many Parameter blocks are added to the model. The output node is labeled based on the name given to the parameter.

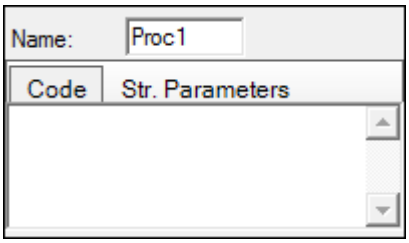
The input node on the Parameter block can be connected to output nodes on PK compartments, observation blocks, PD blocks, and other parameter blocks. Once an output node is connected, its name cannot be changed.

- Insert a Parameter block by selecting **Insert > Parameter** from the right-click menu.
- Use the pointer to select a Parameter block.
- In the **name** field, type a name for the parameter.

Procedure block

Use Procedure blocks to enter assignment statements, differential equations, and logical statements. Within a Procedure block, functions in the Code tab are executed in the order they are listed, one or more times at each simulation step. For that reason, Procedure blocks are not suitable for counting, storing model status across time, or performing computations at specific times during simulation. Code blocks are case sensitive.

- Insert a Procedure block by selecting **Insert > Procedure** from the right-click menu.



Statements can follow each other on a single line or be set on different lines.

```
statement statement …
```

The most common type of statement is an assignment, for example:

```
x=5+y;
```

This statement assigns the variable *x* the value 5+y. Note that *x* and *y* must be valid drug model variables. The semicolon character (;) separating statements is optional.

The statements are inserted in the model, alongside automatically generated statements. For example, users can define new structural parameters Q and R, and then enter this code in the block:

```
deriv(B=-B*Q)
D=B/R
```

It is important not to write code that depends on time "t" or other continuously changing variables to equal exact values, such as:

```
x=((t==4)?xxx:yyy)
```

because this code is executed very frequently, including every time a derivative evaluation is performed. Time "t" takes on many values, along with other continuously changing variables, but cannot be relied on to equal any particular value, or even to advance monotonically. However, covariates, if not interpolated, use the specific values that they are set to by the input data.

Differential equations can be created only by using deriv statements inside a Procedure block. Note that the variable *t*, the time variable, can be used inside of a deriv statement. *t* represents subject time, which is the elapsed time since the first event experienced by the subject.

Also, it is not a good idea to write code containing discontinuities, such as

```
x=((t>=4)?xxx:yyy)
```

because x discontinuously changes from xxx to yyy at time 4. Assuming x enters the model in some way, it can cause the ODE solver or the modeling engine to be faced with a discontinuity. If the ODE solver encounters a discontinuity, it responds by adjusting the step size to as small a value as possible in order to minimize error, which can greatly slow model performance. If the modeling engine encounters a discontinuity, it responds by trying (and failing) to make a continuous model in the region of the discontinuity, resulting in a troublesome fit and possible lack of standard errors. Use the `sequence` statement in the PML to schedule events when discrete actions can be taken.

Discrete covariate values can be referred to using their number representation. Each value is represented by a whole number, starting with 0 (zero) for the first value listed in the Covar. Type tab, which is located in the Parameters tab.

For available logical operators and functions that can be used in statements, see "Supported Operators", "Supported Math Functions", and "Supported Special Functions".

To specify Procedure block options

- Use the pointer to select an Parameter block.
- In the **Name** field, type a name for the procedure or use the default.

  The Procedure block is labeled Proc1, Proc2, and so forth depending on how many Procedure blocks are added to the model. The output node is labeled based on the name given to the parameter.
- In the Code tab, enter any differential equations or PML (Phoenix Modeling Language) statements.
- In the Str. Parameters tab, click **Add** to add a structural parameter to the Procedure block.
- In the parameter field, type a name for the parameter.

  An input node is added to the Procedure block for each new parameter that is added.
- Click the **X** button to remove a structural parameter.

  If the statement entered in the Code tab adds a dose point to the model, then the Dosepoints tab allows users to specify if the dose point uses infusion.
- If the dose point uses infusions, check the **Infusions possible?** checkbox in the Dosepoints tab.

  Selecting this option adds a dosing rate context to the Main Mappings and the Dosing panels.

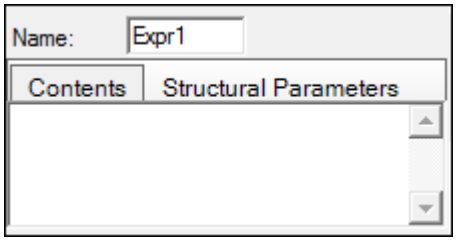
Note: Sequence blocks cannot be entered in a procedure block when working with the graphical model. They can be entered in the textual model, however.

Expression block

The Expression block represents one variable, whose value is set by a text expression entered in the Contents tab. Note that this can only be an expression; no statements, such as assignments (using

the “=” sign), or PML code can be included. An expression is any combination of numbers, identifiers, and operators representing a single value. It includes no “=” sign representing assignment.

- Insert an Expression block by selecting **Insert > Expression** from the right-click menu.



An expression represents a single numerical quantity in algebraic terms. It does not contain an assignment, or “=”.

An expression may include any combination of the following:

Numbers
Optional decimal point: `999 9.99 .999 999. 0.999`
Followed by optional exponent: `1e999 99e-999`
Limits: IEEE double precision (about 13 decimal places)

Identifiers (variable names, function names)
Identifiers used within expressions must be valid model variables or function names.

Unary minus (negative sign)
Syntax: `-expression`
Example: `-exp(-ETA1)`, where `ETA1` is a variable in the model

Factors
Syntax: `expression*expression`
or `expression/expression`

Sums
Syntax: `expression+expression`
or `expression-expression`

Exponential terms
Syntax: `expression^expression`
or `expression**expression`
Example: `Emax*C^gamma/(EC50^gamma+C^gamma)`
where `Emax`, `C`, `EC50` and `gamma` are variables in the model.

Numeric Comparisons
The value of an expression using a comparison is one if the comparison is true, zero if false.
Comparison operators allowed within expressions are:
- greater than or equal to: >=
- less than or equal to: <=
- not equal to: !=
- is equal to: ==
- greater than: >
- less than: <

Syntax: `expression1 == expression2`
Example: `probability_pain_relief >= chance`

Choice operator
Syntax: `expression1 ? expression2:expression3`
Example (conditional expression): `bodyweight < 300 ? 0:1`
If bodyweight is less than 300, the value is zero; else the value is one.

Logical operators
For example: &&, ||, !

These have a lesser level of precedence than arithmetic operators. So, for example, "a+b>c" means "(a+b)>c".

Example expression using a discrete covariate:

The values of a covariate ***baseline_pain*** are low, moderate, and severe, in that order. They are represented internally by the Phoenix as 0, 1, and 2.

An Expression block called "pain_effect" might contain a conditional expression as follows:

```
Baseline_pain > 0 ? -1:0
```

This expression reads, "if baseline pain is greater than zero (the index for the first discrete value, low), the expression value is negative one; else, zero."

The equality operator (==) can also be used to build expressions from discrete covariates or model variables such as:

```
CL=tvCL+(renal_necrosis==1)*renNC
```

For available logical operators and functions that can be used in expressions, see "Supported Operators", "Supported Math Functions", and "Supported Special Functions".

To specify Expression block options

- Use the pointer to select an Expression block.
- In the **Name** field, type a name for the expression or use the default.

  The Expression block is labeled Expr1, Expr2, and so forth depending on how many Expression blocks are added to the model. The output node is labeled based on the name given to the parameter.
- In the Contents tab, type the expression.
- In the Structural Parameters tab, click **Add** to add a structural parameter to the Expression block.
- In the parameter field, type a name for the parameter.

  An input node is added to the Expression block for each new parameter that is added.
- Click the (**X**) button to remove a structural parameter.

Annotation block

The Annotation block is used to add notes to a model.

- Insert an Annotation block by selecting **Insert > Annotation** from the right-click menu.
- Use the pointer to select an Annotation block.
- In the **Note** field, type a note or description.
- Check the **Border** checkbox to add a border to the annotation.

  Annotation blocks do not wrap the text.
- Press ENTER to add another line to the annotation.

  If the Annotation block has a border, the block can be expanded to surround the text.

- Expand the block by selecting it with the pointer.
  Place the pointer over one of the blue squares on the corners of the Annotation block. Single-click and drag the blue square to expand the block around the text.

Vascular flow

Use a vascular flow to represent blood flow in physiologically based PK models. Use this flow to connect PK compartments representing organs or other areas into circuits of blood flow. The flow rate between organs is represented by the variable Q.

- Insert a Vascular flow by selecting **Insert > PBPK > Vascular Flow** from the right-click menu.
- Use the pointer to select the first compartment of the flow.
- Use the pointer to select the second compartment of the flow.

Note: A Vascular Flow should connect a central or a peripheral compartment. It should not be used to connect an elimination or an absorption compartment.

To specify Vascular Flow options

- Use the pointer to select a Vascular Flow.
- Click **Draw** multiple times to toggle through the options for drawing the PK flow in the Graphical model interface.

  Diagonal: Draws the diagonal flow.
  Horizontal First: Draws the flow horizontally from the first compartment.
  Vertical First: Draws the flow vertically from the first compartment.
- In the **CL** field, enter a name for the input variable or use the default.

Textual model interface

To use the textual editor to create the structural model

- Click **Edit as Textual** to use PML (Phoenix Modeling Language) code to create the model.
- In the displayed dialog, click **Yes** to confirm using the textual editor.

When using the textual editor, the **Edit as Textual** button changes to either **Use Builtin** or **Use Graphical**, depending on the mode you were in before switching to the textual editor. Click this button to return to the previous mode.

The model text is displayed in the Model panel. It represents the model as it currently exists and allows users to edit the structural model using PML code. The PML examples illustrate the use of PML to create various models and describes the use of the available statements and functions.

While typing, the text editor will provide hints for adding parameters. Type the parameter and its following comma to see the hint for the next parameter.

When in text model mode, some Maximum Likelihood Models object option tabs change:

The General tab displays PML errors and warnings as model code is entered in the Model panel.

The Parameters tab displays only the parameter names and their corresponding initial values. Bounds and initial values should be entered or modified in the fixef statements in the PML code.

The Input Options tab is where infusion information is specified.

The Model Text tab displays only the column definition. The model code is now only displayed in the Model panel.

Note: When doing a Profile of a Textual model, avoid deleting fixef parameters as this can lead to extra copies of fixed effects appearing in the "Fixed Eff" list.

Run modes

The following run modes are available in Phoenix.
(Only the Simple and Simulation run modes are available for individual models.)

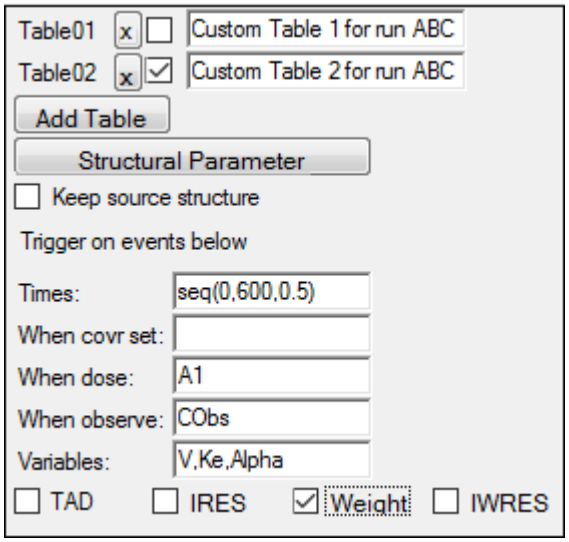
Simple run mode

The Simple run option estimates the parameters for the specified model with the selected options and engine method. The Simple option is available for Population and Individual models and allows specifying additional optional output tables. For example, users can specify an output table whose rows represent instances where particular covariates are set, particular dosepoints receive a dose, or particular observables are observed.

Simple run mode table options

- Click **Add Table** to include extra results tables that contain user-specified output.

  Several options are made available. Users can enter their own values in the new fields to create custom table output that is sorted on a per profile basis.



- Check the checkbox after the table name to view/modify the table settings for that table.
- Click the **X** button to delete the corresponding table.
- Enter a brief description of the custom output table in the field.

  All model variables, including stparms, fixefs, secondary, and freeform assigned variables, can be added as output in the table.
- Click **Structural Parameter** to add the parameters defined in the Structural tab or `stparm` statement to the Variables field.

- Check the **Keep source structure** to keep the number of rows appearing in the table the same as the number of rows in the input datasets. Use the fields below the checkbox to enter the covariates, doses, observations, and variables to include in the table.

  Uncheck this option to allow the rows in the results table to change based on the trigger information entered in the fields below this checkbox.

  - In the **Times** field, enter sampling times to include in the tables. For example, 0.1, 1, 5, 12. A row is generated in the table for each time.

    Additionally, a sequence statement like "seq(0,20,1)" can be used to convey that the times to output in the table should be from zero to 20 by 1 unit of time (e.g., 0, 1, 2, 3, …, 20). The first argument in the seq statement is the starting time point, the second argument is the last time point, and the third argument indicates the time increments.
  - In the **When covr set** field, enter the covariates used, if any. For example, age, weight, gender. A row is generated in the table at any time at which the specified covariate value is set (not missing).
  - In the **When dose** field, enter the dose point to include in the table. For example, Aa or A1. A row is generated in the table at any time at which a dose is administered to the specified compartment, if dosing data is provided in the input source or a dosing worksheet.
  - In the **When observe** field, enter any observations to include in the table. For example, CObs or EObs. A row is generated in the table at any time at which the specified observation is made.
  - In the **Variables** field, enter any additional variables not applicable in the other fields (separated by commas). A column is added to the table for each variable specified in the list.

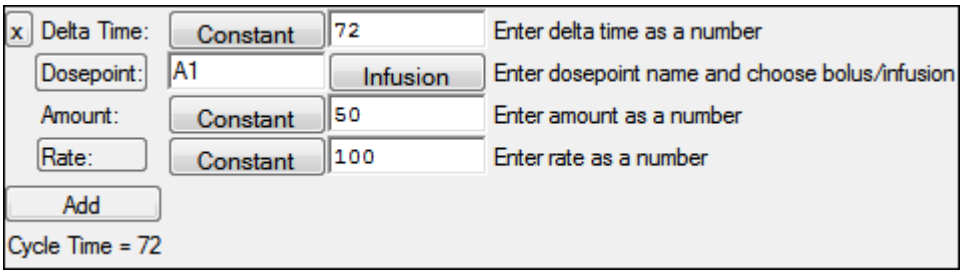
    For example, to capture concentration C and a secondary parameter called C2 at each observation of concentration (CObs) and at time 7.5 enter 7.5 in the **Times** field, CObs in the **When Observe** field, and C and C2 in the **Variables** field.

    The Variables field can also be used to get population prediction (PRED). This is done through a special function "`pred`" with its arguments set to be some observed variable names. For example, to capture PRED corresponding to the observed variables, CObs and EObs, at times 1 and 4 as well as at the observation times for CObs, enter `1, 4` in the **Times** field, `CObs` in the **When Observe** field, and `pred(CObs, EObs)` in the **Variables** field,
- Check the boxes for **TAD**, **IRES**, **Weight**, and **IWRES**, to include these values in a Table. For IRES, Weight, and IWRES, specify an observation for the **When observe** option to make the checkboxes become available.
- The variable names are case sensitive. If the table is not produced as expected, check the case of the variable names in the table specification.

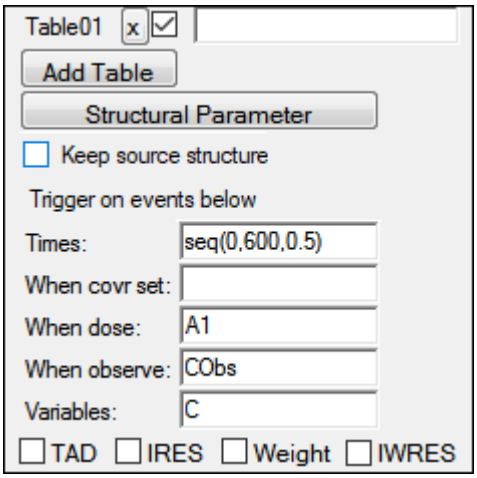
Simple run table use

An example use of the Table option in Simple mode and the Table option is to simulate the effect of adding a number of infusions of 50 dose units infused at a rate of 100 every 72 time units. Select the

ADDL checkbox in the Input Options tab and click **Add**. Then enter values for the dosing regimen as shown below.



In the Run Options tab, the number of iterations, **N Iter**, should be set to zero so that the estimates are fixed. The Simple mode table can be specified to output the times desired to see simulated concentrations. In the image below, the request is to output times from zero to 600 in increments of 0.5. Note that requesting the variable “C” indicates that predicted concentration values (C) should be recorded at the times requested, when doses are added to the amount in the central compartment (A1), and when concentrations are observed (CObs).



Scenarios run mode

The Scenarios run mode option is only available for Population models and it requires that scenarios be defined in the Scenarios tab, which is located in the Parameters tab. Scenarios allow users to investigate different combinations of covariate effects on the same structural model. The Scenarios mode is similar to doing a simple run on several models where their base structural model is the same but the covariate effects are different. If sort keys are specified, then the number of models considered is the product of the total number of scenarios and the total number of unique sort keys.

Covariate Search Stepwise run mode

The Cov. Srch. Stepwise (stepwise covariate search) run mode is only available for Population models. Covariates and covariate effects must be specified. This run mode performs an automatically parallelized stepwise forward or backward addition or deletion of covariates effects by adding one at a time to determine if they make a sufficient threshold improvement based on the specified criterion options. (Note that, during the first step, the baseline model is combined with the cases for the first addition step in order to avoid running the base model alone as the first step.)

Use the **Criterion** menu to select information criterion to use. In the **Add P-Value** and **Remove P-Value** fields, enter the threshold values at which to add or remove a covariate effect.

There are three options for this mode: the criterion on what to base the stepwise approach (**-2LL**, **AIC**, or **BIC**), the threshold for improvement in the criterion in order to add a covariate effect, and the threshold to remove a covariate effect.

If the **-2LL** criterion is chosen, instead of providing threshold values for adding and removing effects, the user provides p-values, such as 0.01 and 0.001. These p-values are used, in conjunction with the degrees of freedom for the particular effect, to determine the thresholds using the inverse cumulative distribution function of the chi-square distribution. The degrees of freedom is the number of fixed effects active under the actual current selection of covariate effects. Normally, this is the base set of fixed effects, plus one for each enabled covariate effect. However, if the covariate effect is for a categorical covariate having N categories (N > 1) the number of fixed effects (and thus degrees of freedom) for that covariate effect is N-1.

The stepwise covariate search method used is the forward addition, backwards elimination where the structural model is used as a baseline and the covariate model is made increasingly complex. After each model estimation, the covariates are evaluated to see which one has the greatest improvement in the goodness-of-fit statistic selected (**-2LL**, **AIC**, or **BIC**) greater than the user-specified threshold. That covariate is added to the regression model for the structural parameter and the model is estimated. This process is repeated until all significant effects are accounted for. Then the process works in reverse to eliminate covariates on parameters whose removal produces the smallest reduction in goodness-of-fit less than the user-specified threshold.This process can find a good set of covariate effects more quickly than the shotgun mode.

The stepwise covariate search option creates a list of models called scenarios, which are listed in the Scenarios tab. The best scenario based on the criterion (**-2LL**, **AIC**, or **BIC**) is selected in the Scenarios tab.

Covariate Stepwise searches only generate the Overall worksheet as an output worksheet. To generate the full results, change the Run Mode to **Scenarios** in the Run Options tab and re-execute. Since the best scenario is selected automatically after the covariate search, it will be used during the Scenarios run.

How to force some covariates to be parts of the model while others are evaluated via the stepwise covariate search

In some Population PK modeling studies, it may be necessary to "force" some covariates to be part of the base model (e.g., Cl=tvCl*(Wt/70)^0.75). However, when it comes to stepwise searching, the rest of the covariates (e.g., age, sex, etc.) need to be tested. This can be accomplished through the text model.

- In the Run Options tab, click **Edit as Text**.
- Look for lines such as:

```
fixef(dVdwgt(enable=c(0))=c(, 0.949657, ))
fixef(dKedGender1(enable=c(1))=c(, 0.0586368, ))
```

- Remove them as desired by deleting or commenting out the enable clause as shown below:

```
fixef(dVdwgt/*(enable=c(0))*/=c(, 0.949657, ))
fixef(dKedGender1(enable=c(1))=c(, 0.0586368, ))
```

The covariate search will ignore the covariate that was commented out (or deleted).

Another way to comment out the enable clause is as follows:

```
/* (enable=c(0)) /**/
```

The /*…*/ comments are non-nesting, so both of these opening and closing character sets would have to be removed in order to comment the clause back in. Using the /*…/**/ characters makes it easier to comment the clause back in:

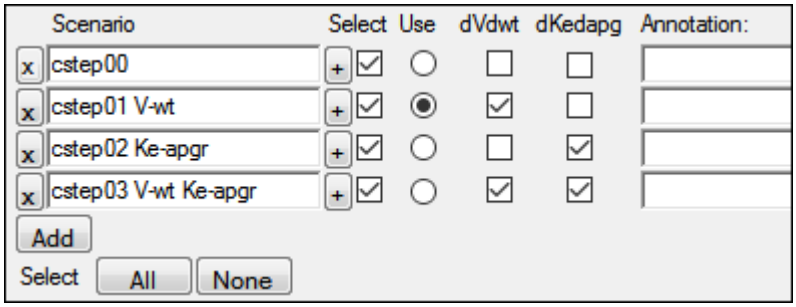
```
(enable=c(0)) /**/
```

Covariate Search Shotgun run mode

The Cov. Srch. Shotgun (shotgun covariate search) run mode option is only available for Population models. Covariates and covariate effects must be specified. This run mode performs a shotgun search to evaluate all the possible scenarios and selects the best one out of 2^(number of effects). Each possible effect that is added has the effect of doubling the number of scenarios. It creates a list of evaluated scenarios in the Scenarios Tab.

Use the **Criterion** menu to select information criterion to use. See “Covariate Search Stepwise run mode” for details on the criterion options.

The best scenario based on the criterion (-2LL, AIC, or BIC) is selected in the Scenarios tab. The selected scenario is used for any subsequent Simple runs. An example of a selected scenario that was created after using the covariate search stepwise mode is shown below.



Covariate Shotgun searches only generate the Overall worksheet as an output worksheet. To generate the full results, change the Run Mode to **Scenarios** in the Run Options tab and re-execute. Since the best scenario is selected automatically after the covariate search, it will be used during the Scenarios run.

Bootstrap run mode

The Bootstrap run mode is only available for Population models. Bootstrap is a diagnostic tool for understanding the precision of estimates. The assumption is that the observations are independent between subjects. Bootstrap consists of running a series of model fits, each time using a random sampling, with replacement, from the original set of individuals.

For example, if a user has 100 subjects in the original study data, then the created datasets, or resamples, will also contain 100 subjects per each bootstrap run, with some subjects replicated more than once. Several bootstrap samples can be created and all of the samples are individually fitted to the original model in order to obtain a new set of estimated parameters for each sample. After all the samples have been fitted, the bootstrapped estimates are summarized and then outputted.

When setting up a bootstrap run, enter the number of samples to create (**# samples**). The number of samples requested can be a number from two to infinity, however, very large numbers, such as one million or higher, are discouraged because of memory limitations.

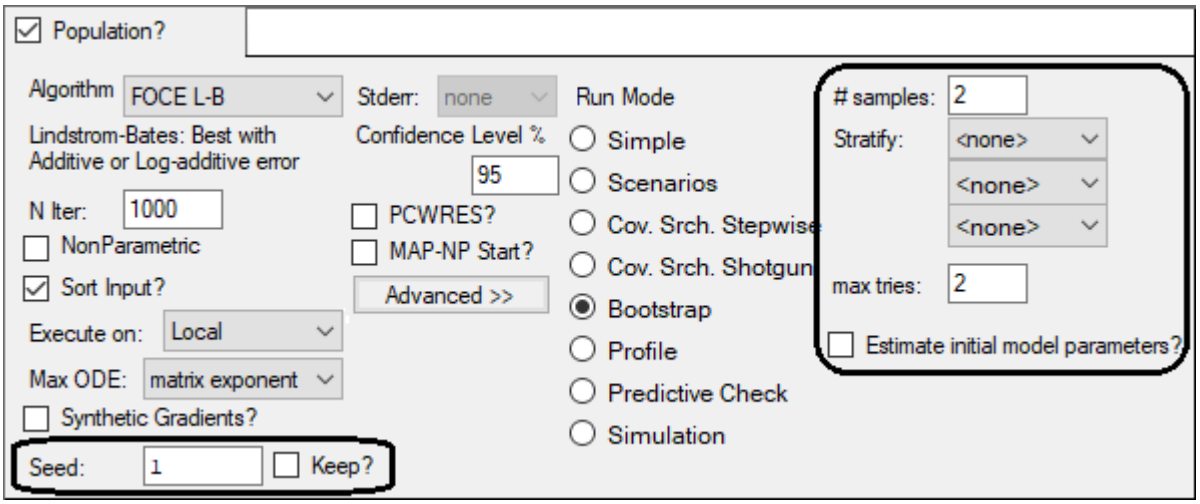
The user can optionally select up to three particular categorical covariates from the **Stratify** menus to stratify the samples. This will ensure that each unique value available for the selected stratification variable will be sampled equally for each Bootstrap run. For example, if a user stratifies the data by

sex and the data has 100 subjects that are 50% male and 50% female, then each resampled dataset will have 50 males and 50 females. A maximum of three stratifications can be selected, and the order of the selected stratified variables specifies a nesting structure. (In cases where reset blocks are involved, only the first values of covariates in the first reset block are used.)

The maximum number of tries (**max tries**) option accepts numbers from 2 to 99. If a sample fails to converge, it is re-tried as many times as indicated with different random seeds, in an effort to get a full set of samples evaluated.

The **Estimate initial model parameters** option, when checked, causes Phoenix to first run a simple fit to the model and then take the final estimates of the simple run as initial estimates for all the bootstrap runs. Typically, this option is not needed, since fine-tuning of the parameter estimates is normally done by the user prior to the bootstrap run and can significantly increase the run time for some models. (***Note***: When checked, the final estimates of the simple run will be copied into the original model text/fields.)

The **Seed** option is in the Run Options tab, and it is used as an initial seed value for the random sampling. Each subsequent bootstrap sample will use a different seed. Phoenix adds 100 to the starting seed. If the **Keep?** option is selected then the same starting seed will be used for the next run. Otherwise, Phoenix assigns a random seed and bootstrap results might vary if they are executed twice.



When running in Bootstrap mode, the **Stderr** option is disabled.

If scenarios are defined in the Scenarios tab using the covariate search methods or by manually creating them, then the scenario selected by the **Use** option button in the Scenarios tab is the one used for the Bootstrap procedure.

In the Fixed Effects tab, the Estimate area will automatically show the final bootstrap parameter estimates (means) after the Bootstrap run of the model is successfully executed.

Profile run mode

The Profile run mode is only available for Population models. This option performs likelihood profiling to help users understand the effect of the fixed effect parameter estimates on the likelihood. This approach perturbs by an indicated amount or percentage one of the selected fixed effects and fixes, or freezes, it while letting the other model parameters vary. On each perturbation, the model is fitted and its results are presented.

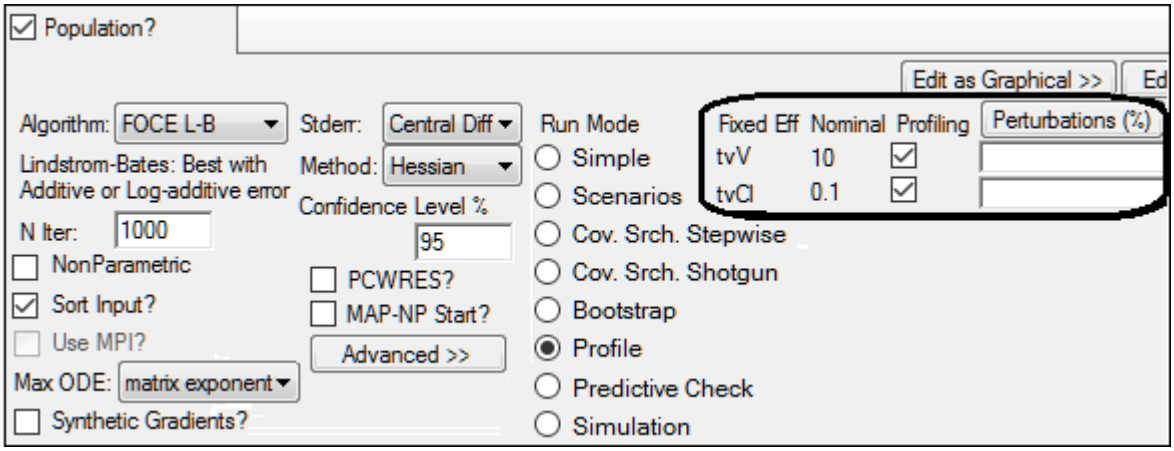
The user can select the fixed effect(s) to investigate by checking the **Profiling** checkbox beside each fixed effect to include. Then enter comma-separated amounts or percentage values by which to per-

turb the parameter estimate in the **Perturbations** field. Click **Perturbations** multiple times to toggle between **Perturbations (%)** and **Perturbations (Amt)**.

If more than one parameter is selected, the profiling is done as many times as parameters are selected. Each parameter is perturbed by the indicated absolute amount or by a percentage of their estimate, and Phoenix will also display the original parameter estimate without perturbation. If a model is very sensitive to a specific parameter, perturbing that parameter by a small amount will show a large effect on the likelihood.

For example, if the initial estimate of tvV is 10 and a user enters the perturbation values as “–1, 1, 3", then the model is run four times with tvV frozen at 9, 10, 11, and 13. If the initial estimate of tvV is 10 and the perturbation percents are “–10,10”, then the models are run with tvV frozen at 9, 10 and 11. A perturbation of zero is the default value and does not need to be entered.

The user selects fixed effects parameters for profiling by placing a selecting the profiling column for the desired parameter. A field is displayed that allows users to enter numeric values separated by a comma (–10, 10). In the Run Options tab, users can switch between perturbations entered by amount or by percentage.



Predictive Check run mode

The Predictive Check run mode is available for the Phoenix Population Model. When the **Predictive Check** run option is selected, the model is first fit with 0 iterations (regardless of **N Iter** in the user interface) in order to generate the data needed to fill the standard results (such as Residuals, Theta, etc.).

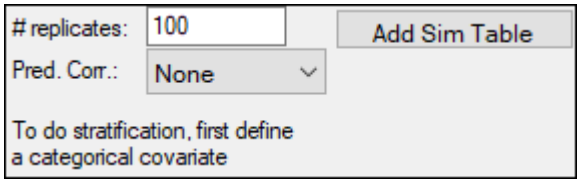
Predictive Check is based on the resulting model fit estimates of Theta, sigma, and Omega. Each simulation replicate generates a dataset just like the original dataset. Both eta (individual variation) and epsilon (observation error) are sampled and used to create a population prediction. Predictive Check provides a series of summary statistics (quantiles) of the simulated data to compare with the observed data.

To run Predictive Check on the final estimates from a model fitting (**Simple** run mode), users should click the **Accept All Fixed + Random** button in the **Parameters > Fixed Effects** sub-tab to accept the final parameter estimates as the initial estimates for the Predictive Check.

This run mode provides the option to enter an initial seed value for the random simulation instead of using a random seed. This option is located in the Run Options tab. Each replicate uses a different seed by adding 100 to the starting seed. If the **Keep** option is checked, then the same starting seed is used for the next run. Otherwise, Phoenix assigns a random seed and predictive check results will vary if the model is run again. The **Keep** option only prevents a new seed from being created before the next run. The number in the **Seed** field is retained from the last run.

The Main tab in the Predictive Check mode provides options applying to all the observed variables. Besides this, each observed variable has a separate tab, and the options provided in each tab depend on whether the observable variable is continuous or discontinuous.

Main tab options



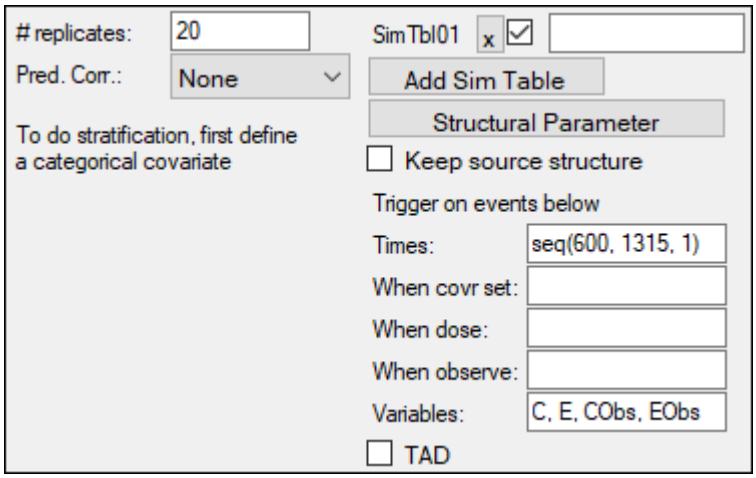
- In the **# replicates** field, enter the desired number of replicates. This is the number of datasets that are sampled to perform the population prediction. The maximum number of replicates allowed is 10,000. If the user enters a larger number, only 10,000 will be generated. The value entered applies to all dependent variables.
- Optionally use the **Pred. Corr.** (Prediction Correction) pull-down menu to specify the type of correction to use to calculate a prediction-corrected observation. This option applies to all dependent variables except for discontinuous observed variables (categorical, count, and time-to-event), where this option is ignored. (See "Prediction and Prediction Variance Corrections" for calculation details.)

  None: do not apply a correction
  Proportional: use the proportional rule.
  Additive. use the additive rule.

  If either **Proportional** or **Additive** are chosen, two additional options are presented:
  Pred. Variance Corr.: If checked, a prediction-variability corrected observation will be calculated and used in the plots.
  Output PRED: If checked, the population prediction results will be included in the PredCheckAll worksheet as a replicate '-1'. If unchecked, the worksheet will only contain information for the replicates.

Note: If the observations do not typically have a lower bound of zero, the **Additive** option may be more appropriate.

- In the **Stratify** menu, select a defined categorical covariate to stratify the modeling simulation, if it is needed.
  This menu is available only when a categorical covariate is defined. Up to three levels of stratification are available and the same stratas are applied for all dependent variables unless they are overridden in the observed variable tab.
- Click **Add Sim Table** to add a simulation table.



Up to five simulation tables can be added. Users can enter unique names for each table in the field next to the checkbox. These names are NOT used in the simulation results. For every simulation table added here, there is a results worksheet generated called Simulation Table 01 (up to 5). The options available for a simulation table are the same as those described in the "Simple run mode table options" section, except for the following two options:

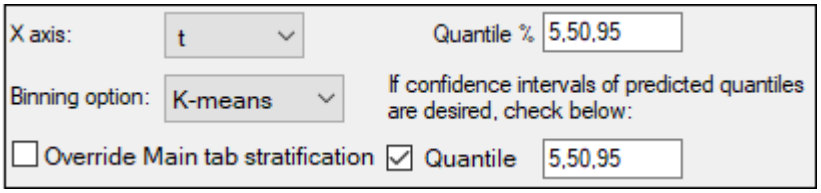
- For the **Keep source structure**, if it is checked, the number of rows outputted in the table for each simulation replicate is the same as the number of rows in the input datasets.
- For the **Variables** field, the user can enter any variables (separated by a comma) used in the model.

For population mode, a worksheet with the results of simulation or a simulation file are loaded into the results: SimulationTable01 or `Rawsimtbl01.csv`. The simulation file is linked as an external file when it is very large and affects performance. The PredCheckAll worksheet contains the predictive check simulated data points. All other results correspond to the model fit because Phoenix fits the model before performing simulations.

See also "Simulation options".

Continuous observed variable tab(s) options

The options provided below are the ones for a continuous observed variable with a model involving categorical covariates.



- Specify the variable to use for the X axis in the predictive check plots from the **X axis** pull-down menu.

  t is time (the default)
  TAD is time after dose
  PRED is population (zero-eta) prediction
  other… displays a field to type in the name of any other variable in the model

  Prediction intervals, or the quantiles of the predictive check simulation, might not be very smooth if there are many time deviations in the dataset. The predictive check has the option whether or not to bin the independent variable.

- Select the binning method to use from the **Binning option** pull-down menu. Different binning methods can be specified for different dependent variables.
  - **None**: Every unique X-axis value is treated as its own bin (the default method)
  - **K-means**: Observations are initially grouped into bins according to their sequence order within each subject. Then the K-means algorithm rearranges the contents of the bins so every observation is in the bin that has the nearest mean X value or "center of gravity." Starting with an initial set of bins containing observations (with a mean X value of these observations), the K-means algorithm:

    1) transfers each observation to the bin with the closest mean to the observation, and
    2) recalculates the mean of each bin. This is repeated until no further observations change bins. Bins that lose all their observations are deleted.
  - **Explicit centers**: Specify a series of X values to use as the center values for each bin. Observations are placed into the nearest bin.
  - **Explicit boundaries**: Specify a list of X value boundaries between the bins. Observations are placed in the nearest bin. The center value of each bin is taken as the average X value of the observations in the bin.

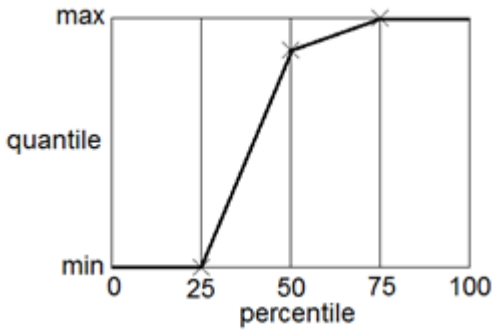
  In the case of **Explicit centers** and **Explicit boundaries**, the numerical values, separated by commas, are automatically sorted into ascending order and duplicates are eliminated. In all cases, bins having no observations are eliminated.
- Check the **Override Main tab stratification** checkbox (available when the model involves categorical covariates) to override stratification rules defined in the Main tab, if it is needed.

  When **Override Main tab stratification** is checked, use the **Stratify** menu to select a defined categorical covariate. The stratas are applied for the current dependent variables only.
- Check the **BQL** checkbox (available when BQL is specified in the model and **None** is chosen in the **Pred. Corr.** pull-down menu) and select an option from the menu to specify the way BQL data is handled. (Only available if the **BQL?** checkbox in the Structure tab is checked or the argument is included in textual mode).
  - Select **Treat BQL as LLOQ** to replace BQL data less than the LLOQ value with the LLOQ value in Observations and related worksheets.
  - Select **BQL Fraction** from the menu to have the amount of BQL data checked and its fraction compared with the quantile level. If the fraction of BQL data is more than the defined quantile, the corresponding observed data are not shown in the VPC plot (Pop PredCheck ObsQ_SimQCI/ Pop PredCheck ObsQ_SimQ) or in PredCheck_ObsQ_SimQCI/ PredCheck_SimQ worksheet. However, the data can be viewed in the BQL fraction plot.
- In the **Quantile %** field, accept the default quantiles or enter new ones.

  Separate multiple quantiles by commas. For example, 10,50,90. The default quantiles are 5, 50, and 95. Different quantiles can be specified for different dependent variables.
- Check the **Quantile** checkbox to include predicted quantiles confidence intervals in the predictive check output. If this option is not checked for a dependent variable, then the associated plot will be the same as the Pop PredCheck ObsQ_SimQ plot.

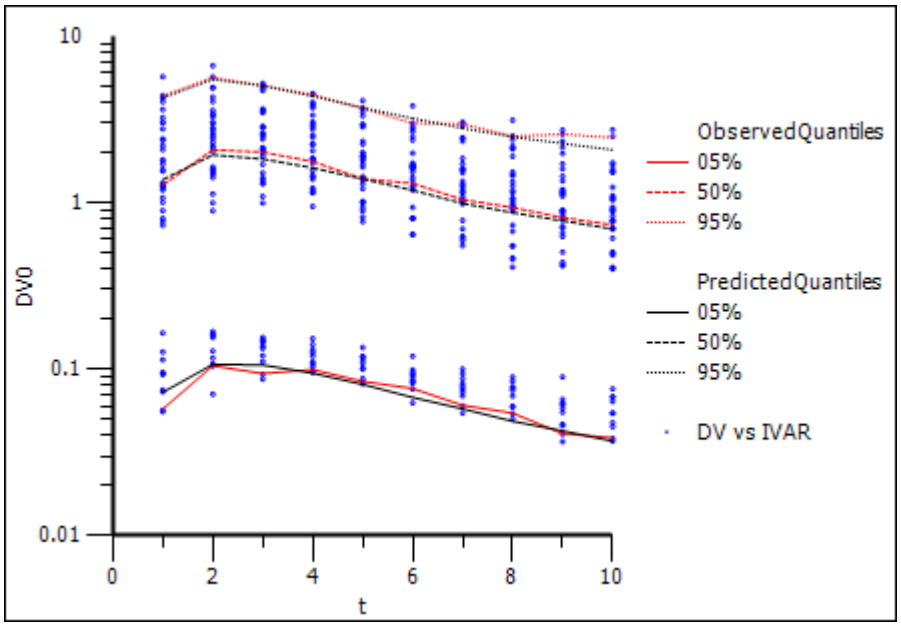
  The quantiles are the summary statistic measures provided by the predictive check. There are multiple ways to calculate quantiles, given percentiles and a data set. NLME uses the same method as SPSS, and R's quantile function with type=6. The following diagram illustrates the quantile function in the case of a dataset containing three numbers. The numbers are sorted in ascending order. If there are N numbers, there are N+1 intervals. Percentiles (expressed as frac-

tion F) falling below 1/(N+1) or above N/(N+1) have a quantile equal to the minimum or maximum, respectively. Otherwise, the quantile is found by linear interpolation.



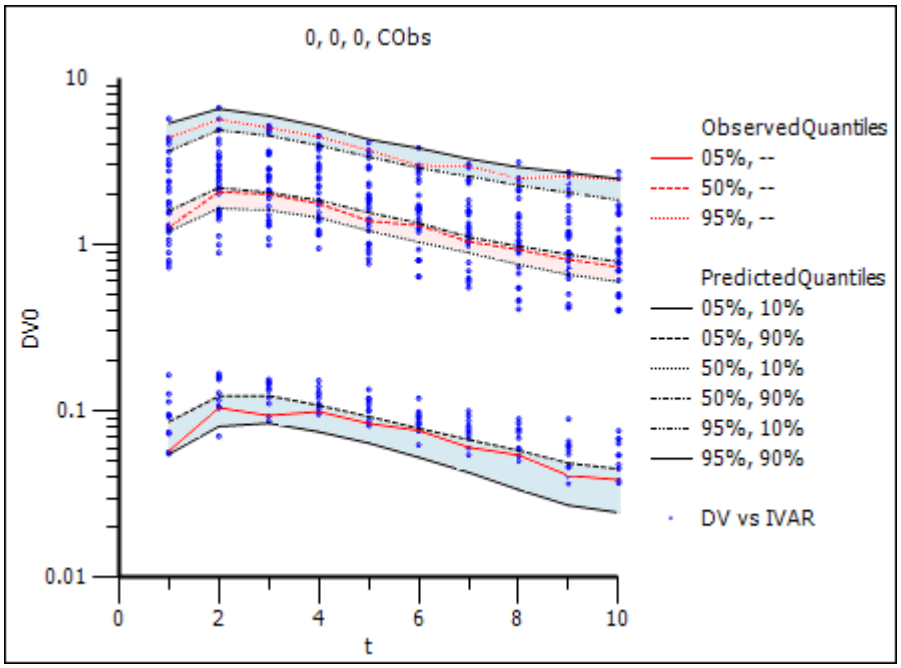
Users can optionally select to calculate confidence intervals for the predicted intervals, or predictive check quantiles. Since each simulated replicate is like the original dataset, first the quantiles are obtained at each stratum-bin-time for each replicate. For each stratum-bin-observed quantile, users get a cloud, one for each replicate. Then quantiles of the quantiles are calculated by stratum-bin-time over all replicates corresponding to a confidence interval of the simulated quantiles.

With predictive check, there is one level of simulated quantiles of observations (e.g., the values entered in the **Quantile %** field). During simulation, the model is purposely disturbed so that the predicted values of the observations fall in a range. The intent is to determine if the range includes the actual observations that came in on the original data. The Pop PredCheck ObsQ_SimQ plot shows the simulated quantiles:



The blue dots are the actual original observations. The red lines are 5%, 50%, and 95% quantiles from the actual original observations. The black lines are the 5%, 50%, and 95% quantiles from the simulated observations. The example plot above shows a good match between them.

Sometimes the user wants to determine the confidence level of the simulated quantiles. To see how much the simulated quantiles are themselves variable, check the **Quantile** checkbox. This provides an additional plot of confidence intervals of the simulated quantiles (Pop PredCheck ObsQ_SimQCI plot).

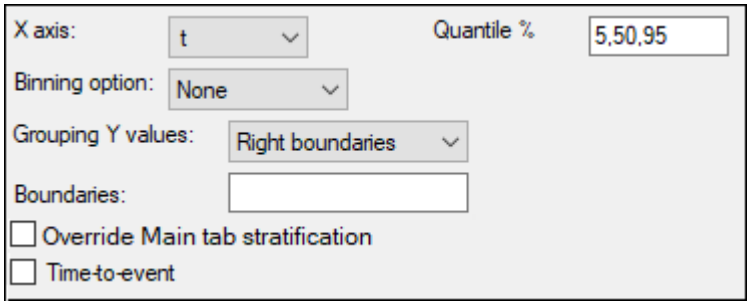


In place of each simulated quantile black line, there are two black lines, representing the 10% and 90% confidence intervals of that quantile. The shading aids in visualizing the variation. Notice how the red lines fall inside the black lines (i.e., within the shaded area), which is a positive result.

Discontinuous observed variable tab(s) options

Many of the options available for a discontinuous variable are the same as those for the continuous variable, Descriptions for those common options can be found in the "Continuous observed variable tab(s) options" section.

Note that the **Xaxis** and **Binning option** are available for categorical and count observations (i.e., it is available only when the **Time-to-event** checkbox is unchecked).



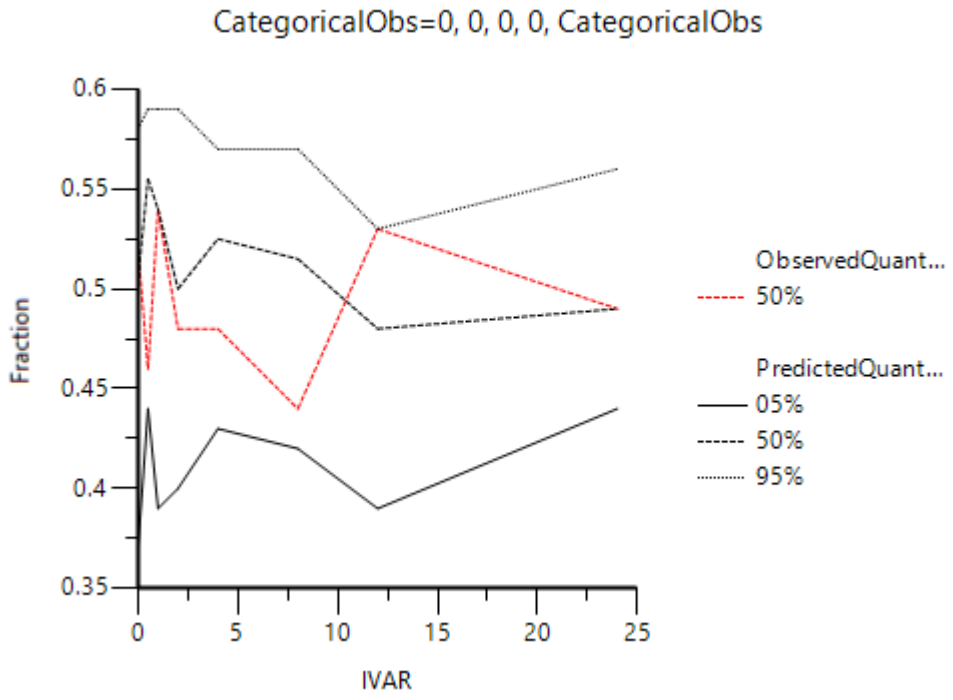
- Use the **Grouping Y values** menu to specify whether Y (dependent variable) values are to be grouped (**Right boundaries**) or not (None) to calculate the summary statistics. This option is available for categorical and count observations (i.e., it is available only when the **Time-to-event** checkbox is unchecked).

  If **Right boundaries** is selected, specify a list of Y values (separated by comma) to define the right boundaries of grouping intervals in the **Boundaries** field. For example, if one enters 1 in the **Boundaries** field, as shown in the image above, then the system will calculate the observed and simulated fraction observations for Y=0 and Y >=1.

- The **Time-to-event** checkbox becomes available in cases of event, count, LL, or multi-statement. When checked, the **Times** field shows the same syntax as in the simulation tab and an additional worksheet is generated (Predcheck_TTE). The user should use this option when Kaplan-Meyer plots are required for a particular variable.

  Note that, in the case of time-to-event simulations, only the first event is used, subsequent events are ignored.

The image below shows the plot of observed fraction (depicted by the red line) and the 5%, 50%, and 95% quantiles (the default values entered in the **Quantile %** field) of simulated fraction (depicted by the black lines) for a categorical observed variable (CategoricalObs) having a value 0 (i.e., CategoricalObs = 0) versus the chosen X-axis.



Simulation run mode

Simulation is useful for situations where modelers have PK/PD parameter estimates from prior modeling and want to explore the potential impact of modifying certain conditions, such as dosing regimens, without having to collect more data. Simulations are based on the structural model and its parameter values.

When the run mode in the Run Options tab is set to **Simulation**, Phoenix performs a PK/PD simulation according to the model settings and PK/PD parameters provided. All engines bypass fitting if **Simulation** is selected (for **Naive pooled**, the variance inflation factor computation is done). Simulation can be performed with built-in, graphical, or textual models.

For built-in and graphical models, users must enter PK/PD parameter values in the **Parameters** tab. Final estimates created by a previous modeling run can be used as initial estimates.

For population mode, a Monte Carlo population PK/PD simulation is performed based on the values provided for Theta, Omega, and sigma. If a table is requested, two separate simulations are performed automatically, one simulation for the predictive check and another simulation for each table.

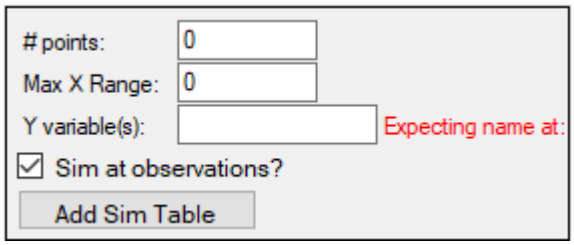
For individual mode, if the input dataset contains multiple subjects, the simulated data will be the same for all subjects unless a parameter values worksheet is mapped to the Parameters panel or the

Use Internal Worksheet checkbox is selected in the Parameters panel and values are entered for each subject. Manually entered parameter names must exactly match the names defined in the model. They are also case sensitive (tvKa, for example). The Simulation run mode creates a Simulation worksheet containing the simulated data points and an Ind Simulation graph containing simulated data.

Simulation options

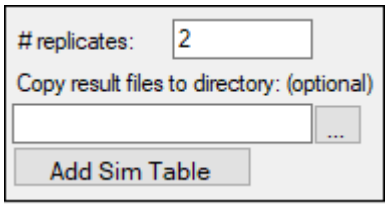
- The following options are specific to the type of simulation model: Individual or Population.

For Individual models (model is only evaluated, not fitted):



- Specify the number of simulated data points to generate in the **# points** field. The maximum number of simulation points allowed is 1,000. The value entered applies to all dependent variables.
- Use the **Max X Range** field to specify the maximum time or independent variable value used in the simulation.
- In the **Y variables** field, specify the desired output variable(s) to capture from the model. The captured variable(s) is displayed, in # points equal time/X increments, in the simulation worksheet. For example, entering 100 in the **# points/replicates** field, 20 in the **Max X Range** field, and C, E in the **Y variable(s)** field simulates a PK/PD model that has 100 data points for C (concentration) and E (effect) between time zero and time 20.
- Check the **Sim at observations?** checkbox to have the simulation occur at the times that observations happen as well as at the given time points.
- Add other simulation tables using the **Add Sim Table** button in the same manner as described under "Simple run mode table options", except that there are no special variables checkboxes available (e.g., **IRES**, **Weight**, **IWRES**).

For Population models, the default tab without any simulation table added will produce the PredCheckAll worksheet only (simulation occurs at the times that the observations happen).



- Specify the number of simulated replicates to be generated in the **# replicates** field. The maximum number of replicates allowed is 10,000.
- If desired, designate a directory for the result files in the **Copy result files to directory** field or use the **[…]** button. If a directory is defined, a csv file with simulation results for all replicates will be placed there.
- Add other simulation tables using the **Add Sim Table** button in the same manner as described under Predictive Check.

Model engines

FOCE Lindstrom-Bates (FOCE L-B)

FOCE L-B applies only to observational data that are continuous and modeled with a Gaussian likelihood. As with FOCE ELS and FO, the random effects (ETAs) are assumed normally distributed as like the residuals (EPSs). Overall this implies that a first-order linearization of the model with respect to ETAs and EPSs will have a normal (Gaussian) distribution of values when evaluated at a given time point for a random individual.

FOCE L-B solves a sequence of linearized mixed effects problems. Each iteration consists of the following steps:

- Conditional step: for each individual, Phoenix finds the optimal ETA values corresponding to the current (THETA, SIGMA, OMEGA) estimates by maximizing the joint log likelihood with respect to the ETAs. This optimization is performed with a quasi-Newton optimization algorithm also used in step 3 as well as other Phoenix engines.
- Linearize the model function with respect to the ETAs around the optimal ETA values computed in step 1. The linearization is used to compute an FOCE approximation of the marginal log likelihood function.
- Solve the linearized mixed effects problem by minimizing the FOCE approximation to the overall negative marginal log likelihood of the linearized problem to obtain a new set of estimates (THETA, SIGMA, OMEGA).

The iterations are repeated until convergence, which is defined by reduction of the gap, or the difference between starting and final optimal log likelihood values for the current linearized problem, to less than a specified tolerance. In Phoenix, the specified tolerance is 0.001. The progress of the current gap value computation is displayed in an external window.

FOCE L-B usually converges, but there is no theoretical guarantee of convergence and both oscillatory and divergent behavior might occasionally occur. The final converged parameter values represent the optimal FOCE solution to the final linearized problem, and are usually, but not necessarily, very close to the optimal FOCE ELS solution.

Note that convergence need not be monotonic. That is, the gaps do not always decrease, and the log likelihoods of the solutions to the linearized problems do not necessarily improve from iteration to iteration.

FOCE ELS

First Order Conditional Estimation-Extended Least Squares, which is applicable to Gaussian data only, is essentially equivalent to the NONMEM FOCE methodology with interaction and is based on minimizing an extended least squares objective function that represents the FOCE approximation to the negative log of the marginal likelihood as a function of (THETA, SIGMA, OMEGA). Unlike FOCE L-B, which requires optimization of approximate marginal log likelihoods for a sequence of linearized

mixed effects problems, FOCE ELS conceptually involves only a single optimization of a top-level approximate marginal log likelihood function.

However, each evaluation of the objective function in the top level minimization requires a conditional step in the form of an inner optimization of the joint log likelihood for each subject with respect to ETA at the current (THETA, SIGMA, OMEGA) values. The inner optimizations, which are the same calculations that are performed in step 1 of FOCE L-B, are nested within the overall optimization process but many more of them are required than in FOCE L-B. Therefore, FOCE ELS is usually considerably slower than FOCE L-B.

The modeling computation window shows the current value of the FOCE marginal log likelihood for the current iteration. Iterations of FOCE L-B are defined in terms of iterations of the underlying quasi-Newton optimization algorithm as applied to the top-level minimization and generally consist of a gradient evaluation followed by a line search along a direction computed from the gradient and previous iterates. Unlike FOCE L-B, progress is monotonic from iteration to iteration in the absence of numerical difficulties that require internal restarts.

Both FOCE L-B and FOCE ELS use "interaction", which means that the individual prediction, which is obtained by using the current optimal ETA estimates to compute the model prediction function, is used to evaluate the residual error model. In contrast, the FO algorithm uses the population prediction, obtained by setting ETA=0, to evaluate the residual error model. The interaction computation is usually regarded as leading to more accurate overall estimates of THETA, SIGMA, and OMEGA.

FO engine

First Order is applicable to Gaussian data only. The FO engine is similar to the FOCE ELS algorithm in that it requires a single top-level minimization of an objective function representing an approximate negative marginal log likelihood. However, the FO objective is simpler and much faster to evaluate than the FOCE ELS objective and does not require any nested conditional optimizations. The FOCE approximation used in FOCE ELS and FOCE L-B requires a model linearization around the current optimal ETAs computed in a conditional step, the FO approximation always linearizes around ETA=0. This means that FO gains speed by omitting all conditional steps, but in doing so sacrifices accuracy. FO is often the fastest and most reliably convergent method, but produces results with the poorest statistical quality in terms of bias and RMSE with respect to true values.

In addition, for purposes of computing statistical weights in the residual error model, FO uses the population prediction obtained with ETA=0. This is less accurate than the individual prediction used by other methods and contributes to the relatively poor statistical quality of FO estimates.

Iterations for FO simply correspond to iterations of the quasi-Newton optimization algorithm. In principle only a single pass through the quasi-Newton method is required, but the Model implementation repeats the optimization from the final results of the previous run until successive runs output the same log likelihood to within a tolerance of 0.001.

Laplacian

Applicable to both Gaussian data and data modeled with general user supplied likelihoods such as count, categorical, and time-to-event data. The Laplacian method has the same basic structure as FOCE ELS in that a top level minimization of an approximation to the marginal negative log likelihood is performed, but the details of the approximation are different and somewhat more computationally complex than FOCE ELS.

The Laplacian engine is based on approximating the marginal log likelihood with a Laplacian approximation to the integral of the joint log likelihood. This replaces the joint log likelihood with a "nearby" Gaussian likelihood called the Gaussian kernel that can be integrated analytically. The determination of the approximating Gaussian function requires both a conditional step as well as a numerical evalu-

ation of the second derivatives of the joint log likelihood function at each point where the top-level objective function is evaluated. This is usually more computationally intensive and less numerically reliable than a FOCE ELS objective function evaluation. Laplacian is often regarded as slightly more accurate but slower and numerically less reliable than FOCE algorithms in Gaussian data cases where FOCE is applicable.

Note: The Laplacian method with FOCE Hess selected is the same as NONMEM's FOCE engine when NAGQ=1. When FOCE Hess is not selected, it is similar to NONMEM's Laplacian engine.

Naive pooled engine

Applicable to Gaussian and user-defined log likelihood data. The Naive pooled engine, when applied to population data, treats all observations as if they came from a single individual in that it ignores inter-individual variations in ETA values. All ETAs are forced to zero, and no OMEGA parameters are computed, only THETA and SIGMA. The engine can also be applied to a single individual, to individuals separately as a series of individual fits in a multiple individual dataset, or to all individuals collectively in a population model. When applied to all individuals in population mode, the engine pools the data for evaluation into a single individual log likelihood function that contains no random effect parameters, but respects inter-individual differences in dosing and covariate values.

The engine minimizes the exact negative log-likelihood, either as a Gaussian or user-specified function. No approximations are necessary since there is no population distribution and hence no joint likelihood to integrate. The same quasi-Newton algorithm as used in the other engines performs the minimization. As with FO, FOCE ELS, and Laplacian, iterations simply correspond to iterations of the quasi-Newton optimization algorithm. Also as with FO, in principle only a single pass through the quasi-Newton method is required, but the Phoenix implementation repeats the optimization from the final value of the previous run until successive runs yield the same log likelihood to within a tolerance of 0.001.

Iterative Two-Stage — Expectation-Maximization (IT2S-EM)

Applies to all types of data, including continuous data, that is modeled with a Gaussian (normal) likelihood, as well as count, categorical, and time-to-event data for which the user must specify a likelihood function). IT2S-EM iteratively performs IT2S and EM-like steps, attempting to improve the approximate marginal likelihood at each iteration. It is not a true EM engine, such as that in MONOLIX. The THETA and SIGMA updates follow an iterative two-stage strategy, while the OMEGA update uses an EM strategy. Additionally, unlike true EM, the ETA estimates are modes of the joint density, whereas EM uses means.

The iteration sequence is as follows:

- Conditional step: for current values of (THETA, SIGMA, OMEGA), for each individual compute an optimal ETA (also known as an empirical Bayesian or POSTHOC estimate). These ETAs maximize the joint likelihood defined by the product of the distribution of the individual residuals, conditioned on ETA, and the population distribution of ETAs. Algorithms such as IT2S-EM, FOCE L-B, FOCE ELS, Laplacian, and Adaptive Gaussian Quadrature all require performance of this same joint likelihood optimization step and are called 'conditional methods' since the evaluation of the approximate marginal likelihood requires computing model predictions that are conditioned on using the ETA values computed by optimizing the joint likelihood.

  In addition, compute covariance (uncertainty) estimates of the ETAs numerically by computing second derivatives of the joint log likelihood function with respect to the ETAs at the optimal ETA values. As a by-product of this computation, the Laplacian approximation to the marginal log likelihood function is obtained for the current (THETA, SIGMA, OMEGA) values.

- Compute new estimates of THETA and SIGMA given the ETAs by optimizing the joint log likelihood function with respect to (THETA, SIGMA) with OMEGA and ETAs frozen at current values.
- Compute new estimates of OMEGA from the ETAs and the uncertainties on the ETAs using the standard EM OMEGA update formula.

The progress of the computation in terms of the value of the current iteration's negative Laplacian log likelihood is displayed in the UI progress bar. Usually the likelihood improves from iteration to iteration, but there is no theoretical guarantee of this happening. The iterations stop based on lack of progress in the log likelihood over several iterations. This can indicate convergence, oscillatory, or even divergent behavior. The best likelihood solution obtained before termination is reported. Since IT2S-EM fit is not an accurate likelihood maximum, standard error results are not reported, as they would also be inaccurate, and possibly not meaningful or even computable.

Often IT2S-EM makes rapid progress during the first few iterations even when the overall sequence of iterates does not converge. A useful strategy regardless of the convergence behavior can be to run a few iterations of IT2S-EM to get an improved starting solution for other engines.

QRPEM

Quasi-Random Parametric Expectation Maximization (QRPEM) is a member of a general class of NLME estimation procedures known as EM methods (in addition to QRPEM in Phoenix NLME, MCPEM from S-ADAPT, SAEM from MONOLIX and NONMEM, and IMP from NONMEM are some of the other currently available EM variants for PK/PD NLME estimation). EM methods for NLME are based on the observation that maximum likelihood estimation usually would become much easier if the true value of the structural parameters were known for each subject. In the simplest case, the maximum likelihood estimate of fixed and random effects parameters can be obtained in a simple single step from the empirical mean and covariance matrix of the known structural parameters.

While the true values of the structural parameters are generally not known, a posterior distribution of the structural parameters (or equivalently, the random effects) can easily be computed for each subject for a given current estimate of the fixed and random effects parameters. Accurately computed means and covariances of these posterior distributions form the basis for a simple computation of updated fixed and random effect estimates. These updated estimates can be shown to have an improved likelihood relative to the starting estimates as long as the posterior means and covariance computations are sufficiently accurate. This procedure can be iterated and will usually converge to the desired maximum likelihood estimates of all parameters.

One major advantage of EM methods is that no formal numerical optimization procedure in necessary to optimize the overall likelihood (or approximation to the likelihood). This contrasts with methods such as FO, FOCE ELS, and LAPLACIAN, which rely on formal numerical optimization using numerical derivatives applied to an approximate likelihood. Numerical optimization procedures, particularly in combination with numerical derivatives, are relatively fragile and can easily fail or be unstable. In contrast, the EM procedures do not rely on numerical derivatives and numerical optimization but rather numerical integration to obtain the means and covariances of the posteriors. Numerical integration is inherently much more stable and reliable than numerical differentiation and optimization.

A second advantage of EM methods is that they may be made as accurate as desired (i.e., they can produce estimates arbitrarily close to the true maximum likelihood estimate). This is done by simply increasing the accuracy of the numerical integration procedure, typically by increasing the number of points at which the integrand is sampled. This contrasts with FO, FOCE ELS, and LAPLACIAN, which are inherently limited in accuracy by the particular likelihood approximations they employ, and which may produce results quite different from the true maximum likelihood estimates.

The key step in most EM methods is computing the means and covariances of the posterior distributions. One common approach is to use Monte Carlo (MC) sampling of the posteriors, assisted by importance sampling techniques. In this case, the method is usually called MCPEM (or in the case on

NONMEM, IMP). Each sample is drawn from a convenient importance sampling distribution such as a multivariate normal that approximates the target posterior, and then each sample is weighted by the likelihood ratio of the target distribution to the importance sampling distribution, evaluated at that sample. The means and covariances of the weighted samples are then used as approximations to the desired true means and covariances of the posteriors. Accuracy can be improved by simply taking more samples.

QRPEM is very similar to importance sampling-based MCPEM, with the exception that samples are no longer randomly drawn but rather taken from a low discrepancy quasi-random sequence (QRPEM uses a Sobol sequence, with an option for Owen or TF scrambling). For purposes of numerical integration, quasi-random sequences fill the domain of interest much more uniformly than random sequences, and usually provide far more accurate integral values for a given sample size.

Many models contain features such as non-linear covariate models, fixed effects not paired with a random effect in a structural parameter definition, or certain types of residual error model parameters. Such models require an auxiliary estimation procedure to obtain estimates for the fixed effects associated with these features. Generally, this involves solving a simple but potentially quite large likelihood optimization model in those parameters, where each sample from each subject contributes a term. This can result in an unnecessarily computationally intensive problem involving a very large number of terms. In these cases, QRPEM applies a resampling procedure called SIR (Sampling-Importance-Resampling) to prune the terms to a much smaller and more manageable number but in a theoretically valid manner. This greatly accelerates this auxiliary procedure without significant loss of accuracy.

Like the other accurate likelihood method AGQ in Phoenix, QRPEM can be applied to both Gaussian and user supplied log likelihood models. As the number of random effects increases, QRPEM becomes increasingly faster and more stable relative to AGQ. However, unlike AGQ, there are two current limitations to the types of models that can be handled directly with QRPEM. The first limitation is that only linear covariate models in a structural parameter definition are allowed (the covariate model must be linear either before or after a log transform of the structural parameter definition). However, in general, models with nonlinear covariate models can be easily handled in QRPEM by a simple manual restructuring of the text model so the non-linear covariate model is applied outside of the initial structural parameter definition rather than inside. Second, if any covariate appearing in a covariate effect has, for any subject, more than one value, such as if it is a time-varying covariate, QRPEM will not run. If a model of either unhandled type is encountered, QRPEM will immediately stop with an error message.

See also the “Structural parameters and QRPEM PML example”.

Adaptive Gaussian quadrature

Adaptive Gaussian quadrature is a generalization of the Laplacian method, and when FOCE Hess is selected the FOCE ELS method, that uses numerical integration techniques to improve the likelihood approximation. It is applicable to both Gaussian and user-supplied log likelihood cases.

When Laplacian with FOCE Hess is selected with NAGQ=1, the resulting method is the same as FOCE ELS and very similar to NONMEM's FOCE engine with interaction. When FOCE Hess is not selected and NAGQ=1, it is similar to NONMEM's Laplacian engine. When NAGQ is set to a value greater than 1, the method has no NONMEM equivalent, and the quality of the likelihood approximation is improved over simple Laplacian or FOCE ELS.

The main difference is that the Laplacian approximation is replaced by a numerical integration step that samples the joint log likelihood function at a grid of ETA values in addition to the ETAs at the maximum. The initial steps are identical, a conditional step to get optimal ETAs followed by numerical second derivative evaluation at the optimal ETAs to compute a Gaussian kernel. In the simpler Laplacian case, the approximation is computed by using only using the joint likelihood value at the optimal ETA

values in addition to the second derivative values, whereas AGQ continues with additional evaluations at special Gaussian quadrature nodes to improve the accuracy of the approximation.

The general AGQ method allows the user to specify the number N of nodes along each ETA variable axis, with the accuracy of the approximation as well as the computational cost increasing with Nd, where d is the number of random effects ETA for each individual. Because of this, AGQ is often most useful for improving estimation accuracy for models with small numbers d of random effects, as is often the case with user-supplied log likelihood functions, particularly for count data.

In the special case of Gaussian data, the user can optionally specify the use of a FOCE approximation to compute the Gaussian kernel covariance matrix instead of numerical second derivatives. This is more stable than a numerical second derivative computation.

NonParametric engine

Applicable to Gaussian or user-defined likelihood data. The nonparametric engine is intended as a post-processor after one of the parametric engines has been run. It makes no assumptions regarding the random effects distribution, conceptually modeling the distribution as a discrete distribution on an arbitrarily fine grid in random effects space. It can be used, for example, to detect bimodality in a parameter such as a clearance. In the nonparametric log likelihood function, the parameters to be fit are the probabilities associated with each grid point in the random effects space.

If the grid is very fine, there can be an enormous number of these probabilities. However, mathematically it can be shown that at the maximum likelihood distribution, almost all of the probabilities are zero, which can be used to simplify the computation. The optimal nonparametric distribution takes the form of a discrete distribution on at most N support points, that is, at most N of the probabilities are non-zero, regardless of how many starting grid points were used.

An iteration for the nonparametric engine involves:

- Selection of a set of candidate support points, which usually includes all of the support points with non-zero probability from the previous iteration plus generation of some additional candidates that are likely to improve the likelihood of the nonparametric distribution.
- Computation of the optimal probabilities on the candidate support points.

The user specifies the number of iterations to apply. On the first iteration, the support points are set at the optimal post-hoc estimates from the initial parametric run. Any fixed effects associated with the residual error model or covariate models are frozen to the values from the parametric run for all of the iterations. A specially designed convex primal dual optimization engine then computes optimal probabilities on these support points. The results of the first iteration are in principle the same as the results of the nonparametric algorithm in NONMEM. However, NONMEM cannot perform any additional iterations, and the final NONMEM support points are fixed at the parametric POSTHOC values, which can be highly suboptimal.

If subsequent iterations are desired, Phoenix first discards any current iteration support points with a probability of zero, and then introduces additional candidate support points and the primal dual algorithm is reapplied to compute a new discrete distribution, which in general will include at least some of the new candidate support points. From iteration to iteration, the likelihood improves monotonically, and the support points migrate to optimal positions. The Phoenix algorithm has the capability of optimizing both probabilities and support point positions using multiple iterations. The NONMEM nonparametric algorithm can only perform a single pass that optimizes probabilities on support points fixed at POSTHOC estimates from a preceding parametric run.

The primary raw result of the Phoenix nonparametric algorithm is the optimal discrete distribution of ETAs in terms of support points and associated probabilities. The means, any covariances, and marginal distributions of each ETA of this distribution are reported. In addition to the optimal population distribution, each individual has a discrete posterior distribution from which a mean ETA value can be

computed. Tables of nonparametric ETA means are produced, as are covariate vs. nonparametric ETA mean plots, which can be used to screen for potential covariate relationships.

Note: If the Model engine gives an exception, it is a general exception caused by a bad fit to data. Should an exception occur, try reconsidering the engine, initial parameters estimates, and number of compartments.

Job control for parallel and remote execution

The following topics are discussed:

Users have the option to execute Phoenix NLME jobs remotely using NLME Job Control System (JCS) or Phoenix JMS. Below is a comparison of the two options:

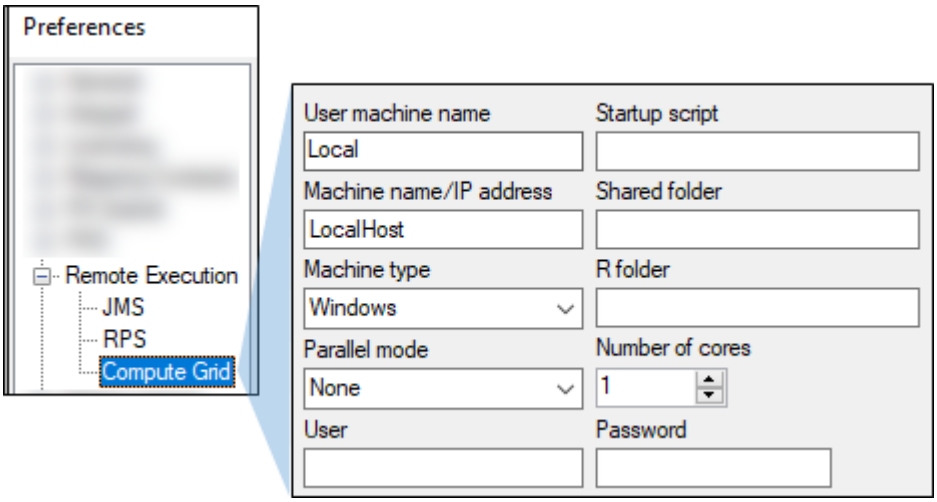
- Remote Windows submission – JMS only
- Remote Linux submission – JCS only
- Disconnect/reconnect/stop – JCS and JMS
- Progress reporting – JCS and JMS
- Remote software required
  JMS – Full Phoenix installation
  JCS –
  - GCC
  - R (batchtools, XML, reshape, Certara.NLME8)
  - ssh
  - MPI (Open MPI for Linux platforms, MPICH for Windows)
- Parallelization method
  JMS – Local MPI
  JCS –
  - MPI for within job parallelization
  - Linux Grid (TORQUE, SGE, LSF)* or MultiCore for between job parallelization

  *TORQUE = Terascale Open source Resource and QUEue Manager, SGE = Sun Grid Engine, LSF = Platform Load Sharing Facility.

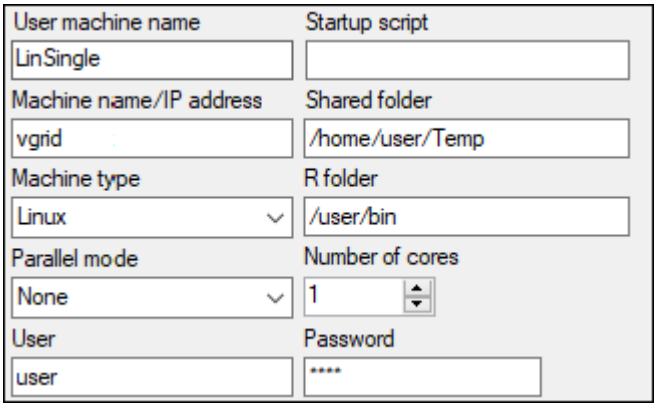
This section focuses on the Job Control Setup, for information on JMS, refer to “Job Management System (JMS)”. Phoenix NLME jobs can be executed on a number of different platform setups, enabling the program to take full advantage of available computing resources. All of the run modes can be executed locally as well as remotely.

One NLME job can be executed using:

- Single core on the local host
  The default configuration profile is as follows:



- Single core on a remote host
  An example of the configuration profile is as follows:

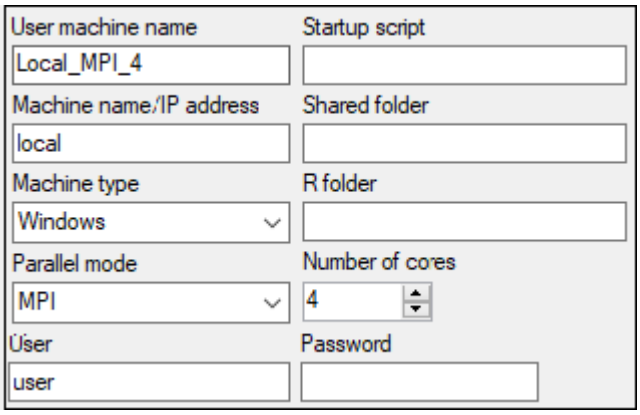


- Using By Subject (first order engines, IT2S-EM)/By Sample (QRPEM) MPI parallelization on the local host

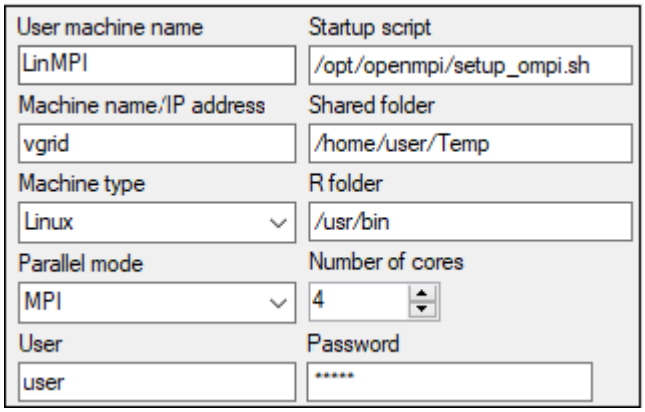
  Subject/Sample method of MPI parallelization is chosen automatically based on the engine chosen. MPI software is required and, for remote host, R and Certara.NLME8 packages should be installed.

  Each model and dataset are unique and the analyst needs to explore the best solution for the current project. However, there are some general guidelines that can be applied to the majority of projects. The speed of model execution is generally based on the number of computational cores, the speed of those computational cores, and the speed of writing to disk. In general, it is thought that an increase in number of cores will result in a decrease in computation time. Thus, parallelizing by 8 MPI threads will be 2x faster than 4 threads. This is true, but the relationship is not linear, due to existing overhead of some unparalleled segments and the overhead of collecting results from different threads.

- For Windows platforms, the default profile with MPI parallelization is as follows:



- For Linux remote runs, an example profile for MPI parallelization is as follows:



Execution can be parallelized by job level for the following run modes:

Simple (Sorted datasets)
Scenarios
Bootstrap
Stepwise Covariate search
Shotgun Covariate search
Profile

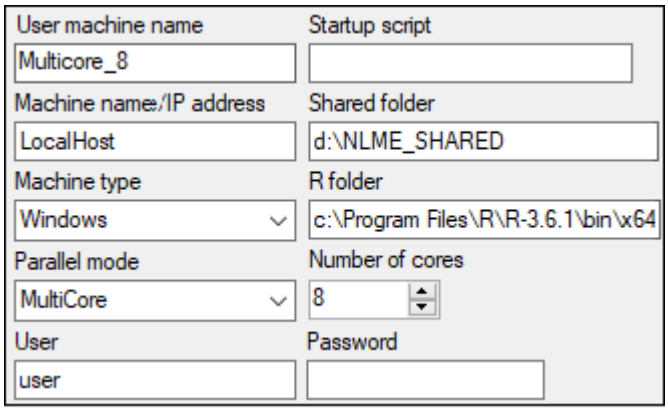
The implemented methods of “by job” parallelization are:

- ***Multicore***: multiple jobs are executed in parallel on the local or remote host

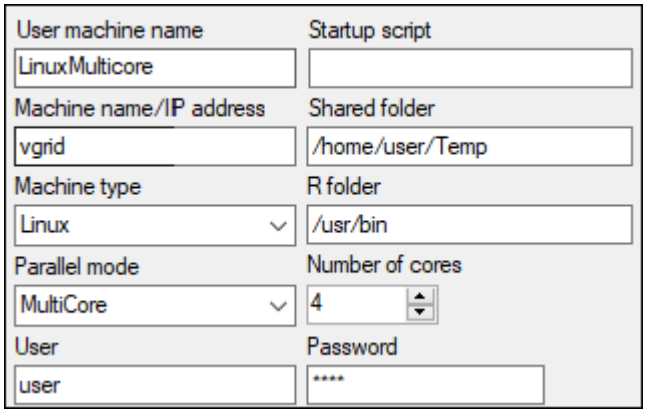
  R and Certara.NLME8 package should be installed on the chosen host. Note that the Multicore method can not be used within MPI (by Subject/by Sample MPI parallelization within each job).

  The user can control the number of processes run in parallel (**Number of cores** field).

  - An example of Windows local configuration is as follows.



- An example of Linux remote configuration is as follows:

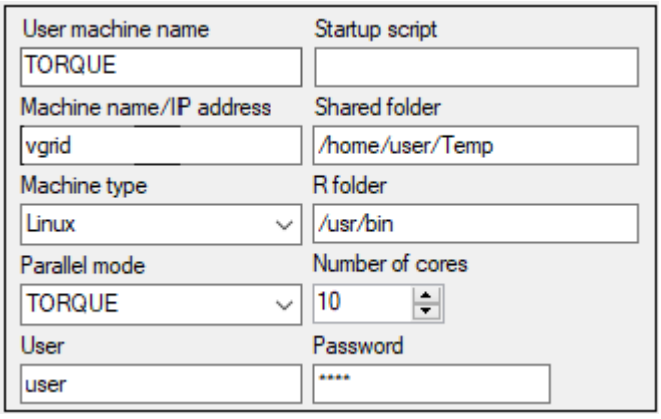


- Submission to a supported remote Linux grid

  Supported grids are SGE, LSF and TORQUE. The number of cores in the configuration for the grid means the number of nodes to be used.

  Number of cores in a grid configuration is the number of cores to be used. Each host can be configured to have multiple cores and each core can handle a separate job.

  An example profile for submission to the TORQUE grid is as follows:



Caution: In some grid configurations, if the number of available cores specified for a grid exceeds the total number of available cores, it can cause the job to remain in the queue. If the job cannot be canceled from within Phoenix, then a direct cancellation through ssh is required. Care must be taken especially for burstable grids, where additional resources (slots) can be requested but not used. Periodic monitoring of the running jobs for the current user is recommended.

The NLME jobs submitted to the grid can be parallelized using MPI if the system has the appropriate MPI service installed and the Parallel mode is set to one of the three ***_MPI** options (**LSF_MPI**, **SGE_MPI**, or **TORQUE_MPI** (to parallelize the runs as by job as well as by Sample/Subject within each job).

For any of the ***_MPI** modes, the number of cores to be used for each job in parallelization will be calculated as the smallest of the following 2 numbers:

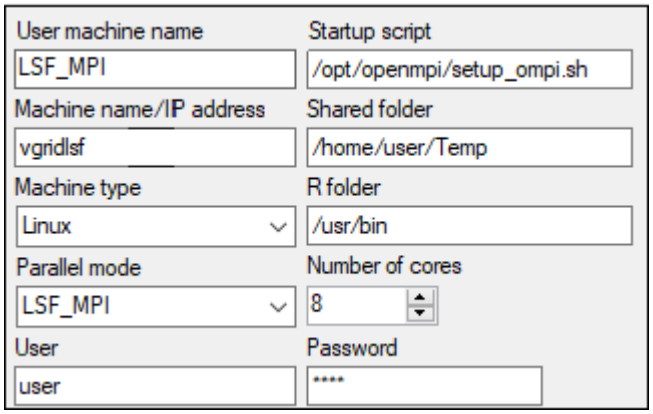
(1) the number of cores in the configuration divided by the number of jobs, or

(2) the number of unique subjects in a specific job divided by 3. If there is an uneven number of unique subjects in each replicate, the smallest number of subjects will be used for the calculation.

Example 1: There are 300 cores available, according to the configuration profile, 4 jobs requested (replicates), and 200 subjects in each replicate. Each of the 4 replicates would parallelize across 66 cores (300/4 = 75. 200/3 = 66. 66 < 75). Total cores used = 264.

Example 2: There are 100 cores available, according to configuration profile, 3 jobs requested (replicates), and 300 subjects in each replicate. Each of the 3 replicates would parallelize across 33 cores (100/3 = 33. 300/3 = 100. 33 < 100). Total cores used = 99.

An example of the configuration profile is as follows:



Caution: For some grid configurations, the number of calculated MPI cores for the particular job cannot exceed the total number of hosts available on the grid. This can cause the software to ask for more hosts to do the computation than are available and result in the job freezing or exiting with an error. In such cases, it is advised to switch to the grid mode without MPI.

Installing job control files

Additional software and libraries are required for certain platform setups.

For within a job parallelization on the local host, the MPICH2 1.4.1 software package is required and is installed during a complete installation of Phoenix or can be selected during a custom Phoenix installation. This application facilitates message-passing for distributed-memory applications used in parallel computing. (If needed, the `mpich2-1.4.1p1-win-x86-64.msi` file is located in

<Phoenix_install_dir>\Redistributables). MPICH2 needs to be installed on all Windows machines included in the MPI-ring or used to submit a job to the MPI-ring. If you have another MPICH service running, you must disable it. Currently, MPI is only supported on Windows.

For parallel processing on a remote host or for using multicore parallelization locally, the following must be installed:

- R 3.3 (or later) can be obtained from http://cran.r-project.org.
- R package 'batchtools' handles submission of NLME replicates to a Linux/Windows GRID.
- R package 'reshape' is used for summarizing the NLME spreadsheet.
- R package 'XML' is used to manage reading/writing `progress.xml`.
- Certara.NLME8_1.0.1.tar.gz is the R package for NLME distributed with Phoenix, located in <Phoenix_install_dir>\application\lib\NLME\Executables. From the R user interface, type the following command:
  ```
  install.packages("Certara.NLME8_1.0.1.tar.gz",repos=NULL,type="source")
  ```
  If the current version of the package is intended to be the default version along the grid, install it with elevated privileges.
- SSH/Sftp needs to be installed on all remote hosts that will be used with the MultiCore parallelization method or act as a submission host to a GRID. SSH is installed by default on Red Hat Enterprise Linux (RHEL) version 7.x. To facilitate job submission from Windows to Linux, NLME in Phoenix (on Windows) contains an SSH library.
- For grid execution (SGE/LSF/TORQUE), installation of Openmpi (v 1.10.6) as a parallel platform on the grid is recommended. Please refer to https://www.open-mpi.org/faq/ for information. Standard grids have Openmpi installed by default.

MPI configuration

MPI configuration is done by the Phoenix installer at the time of installation.

- To check MPI operation, see if there is an SMPD service running on Windows by entering the following into a command shell:
  ```
  smpd –status
  ```
- If it is reported that no SMPD is running, enter the following command:
  ```
  smpd –install
  ```
- Running `smpd –status` again should report that SMPD is running on the local Windows host.

Note: If one of the MPI processes crashes, other MPI processes and the parent process `mpiexec.exe` may be left running, putting Phoenix NLME in an unpredictable state. Use the Task Manager to check if `mpiexec.exe` is still running and stop it, if it is. This will stop any other MPI processes.

Phoenix Model job status

The *NLME Job Status* window is automatically displayed when a Phoenix Model object is executed or by selecting **Window > View NLME Jobs** from the menu.

The window provides easy monitoring of all NLME jobs that are executing as well as a history of jobs that have completed during the current Phoenix session.

During local execution of a Phoenix model, the status window displays the parameter values and gradients of each major iteration of the algorithm, with the most recent iteration at the top. The model name, protocol, stage, time that the execution started and stopped, the status of the job, number of runs, and the number of runs that completed or failed are listed at the top. (See “Job control for parallel and remote execution” for more execution options.)

The job can be canceled (no results are saved) or stopped early (results obtained up to the time when the job is ended are saved) by right-clicking **Status** menu. Specifically, if the **Stop early** button is clicked during an iteration, the run is ended once the current iteration is completed. If the **Stop early** button is pushed during the standard error calculation step, then Phoenix stops the run and prepares outputs without standard error results.

-	Model	Protocol	Stage	Start	End	Status	Num	Completed	Failed
	Diagonal Model	SingleCore		12:02 PM	12:02 PM	In progress	1	1	0

Iteration	-2LL	tvKa	tvV	tvCl
18	358.586	1.586	0.460099	0.0400251
17	358.586	1.5851	0.460389	0.0400685
16	358.602	1.58423	0.458812	0.0402111
15	358.65	1.57986	0.460139	0.0395273
14	359.082	1.57628	0.474117	0.040526
13	359.394	1.5863	0.453124	0.0416298
12	359.416	1.57599	0.456093	0.0372618
11	359.882	1.56582	0.458404	0.0421813
10	360.708	1.56933	0.475879	0.0426218
9	366.256	1.57426	0.434155	0.0332002
8	368.102	1.59489	0.459976	0.0512557
7	368.938	1.6105	0.465461	0.0504036

When **Stop Early** is executed, the Overall results worksheet will show the Return Code of 6, indicating that the fitting was not allowed to run to convergence.

Next to the name of the model that is executed in a run, several pieces of information are shown, providing a quick summary of the job’s current state. The Status of a job can be **In Progress**, **Finished**, or **Canceled**.

If a job involves multiple iterations, information for each completed iteration is added as a row. Use the button at the beginning of the job row to expand/collapse the iteration rows.

Right-clicking a row displays a menu from which the job can be stopped early (the results up to the last completed iteration are saved) or canceled (no results are saved).

Model output

The Maximum Likelihood Models object generates a rather large amount of output:

Worksheet output

Phoenix model worksheet output depends on the run mode selected and whether the model is population or individual.

BootOmega (Population; Bootstrap): Mean estimated Omega matrix over all sample runs. Includes the mean of the covariance matrix of the random effects and the mean Correlation matrix. Bootstrap scenario results will have a suffix of '(B)'.

BootOmegaStderr (Population; Bootstrap): Estimated standard errors for the Omega matrix over all sample runs. The standard errors are derived from the variance of each omega element over all bootstrap sample runs. Bootstrap scenario results will have a suffix of '(B)'.

BootOverall (Population; Bootstrap): Reports the replicate number, the model return code, as well as the likelihood (LL) for all bootstrap runs. The table includes runs that were unsuccessful. Bootstrap scenario results will have a suffix of '(B)'.

BootSecondary (Population; Bootstrap): Reports secondary parameters over all replicates of a bootstrapped population model. The mean, standard error, CV%, median and 2.5% and 97.5% percentiles are included. Bootstrap scenario results will have a suffix of '(B)'.

BootTheta (Population; Bootstrap): Reports fixed effects over all replicates. Includes the mean, standard error, CV%, median and 2.5% and 97.5% percentiles. Bootstrap scenario results will have a suffix of '(B)'.

BootVarCovar (Population; Bootstrap): Reports variance-covariance matrix for fixed effects over all replicates. Bootstrap scenario results will have a suffix of '(B)'.

ConvergenceData (Population; Simple, Scenarios): Lists values for each model parameter at each iteration.

Doses (Individual/Population; Simple, Scenarios, Predictive Check, Simulation): Reports dosing information for each individual used in the model.

Eta (Population; Simple, Scenarios, Predictive Check, Simulation): Reports post hoc or EBE (empirical Bayesian estimates) of the random effects (eta) for each individual.

EtaCov (Population; Simple, Scenarios, Predictive Check, Simulation): Same information as EtaCovariate and EtaCovariateCat worksheets, but covariates appear as columns so there is one row per subject.

EtaCovariate (Population; Simple, Scenarios, Predictive Check, Simulation): Table by subject of individual continuous covariates and individual post-hoc or EBE of the random effects values.
This worksheet is used to create scatter plots to visualize potential covariate effects.

EtaCovariateCat (Population; Simple, Scenarios, Predictive Check, Simulation): Table by subject of individual categorical covariates and individual post-hoc or EBE of the random effects values.
This worksheet is used to create box-plots to visualize potential covariate effects.

EtaEta (Population; Simple, Scenarios, Predictive Check, Simulation): Same information as the Eta worksheet, but presented in a different format to facilitate plots.

Initial Estimates (Individual/Population; Simple, Scenarios, Bootstrap, Predictive Check, Simulation): Reports the initial estimates used.

NonParEta (Population; Simple, Scenarios, Predictive Check, Simulation): Reports post hoc random effects estimates for each subject when the NonParametric Method is selected. This eta vector for each subject is estimated from the nonparametric algorithm, not from the original fit.

NonParOverall (Population; Simple, Scenarios, Predictive Check, Simulation): Reports post hoc random effects mean and variance/covariance estimates for the population when the NonParametric Method is selected. These are computed directly as the mean and variance/covariance matrix of the nonparametric distribution defined by the nonparametric support points and associated probabilities. In the parametric case, the etas are assumed to have a normal distribution with mean zero and a variance/covariance matrix Omega. A nonparametric mean that is significantly different than zero, or a nonparametric Omega matrix that is significantly different than the parametric Omega computed under the normality assumption, provides evidence that challenges the normality assumption. This table is created only when the NonParametric Method is selected.

NonParSupport (Population; Simple, Scenarios, Predictive Check, Simulation): Reports a set of support points and their probabilities, which add up to 1, that define the discrete nonparametric distribution obtained by the NonParametric Method. Unlike the post hoc eta values found by the parametric method under the normality assumption, the nonparametric eta support vectors do not have a one to one correspondence with specific subjects. The number of supports is determined by the fitting algorithm and never exceeds the number of subjects (it is often considerably less). While the fitted nonparametric distribution is discrete, conceptually the true eta distribution is usually regarded as continuous. The support point positions and associated probabilities in the discrete distribution give an indication of where the most likely regions of the true eta distribution lie. This table is created only when the NonParametric Method is selected.

Observations (Population; Predictive Check): Reports the time elapsed and value for each continuous observation. Available for Predictive check only.

Observations_Categorical (Population; Predictive Check): Reports the time elapsed and value for each categorical and/or count observation. Available for Predictive check only.

Omega (Population; Simple, Scenarios, Predictive Check, Simulation): The estimated Omega matrix, the covariance matrix of the random effects multivariate normal distribution, correlation and summary eta shrinkage statistics. Eta shrinkage data is computed both on an overall summary basis (i.e., for each eta_j a summary shrinkage is computed for all subjects) and on an individual subject basis (i.e., for each subject). Refer to "Shrinkage calculation" for details. The summary eta shrinkage statistics also appear at the end of the Core Status file.
The individual shrinkage statistics are reported in the `dmp.txt` file, as well as in the output text file `bluptable.dat`.

OmegaStderr (Population; Simple, Scenarios, Predictive Check, Simulation): The estimated standard errors of the Omega estimators.

Overall (Individual/Population; Simple, Scenarios, Stepwise, Shotgun, Predictive Check, Simulation): Reports model fit diagnostics. Lists the following columns of information:

RetCode: Return code indicating the success of the convergence

LogLik: The log likelihood

-2LL: twice the negative log likelihood

AIC: Akaike information Criterion, a goodness of fit measure based on the log likelihood adjusted for the number of parameters (degrees of freedom) in the fit. It is computed as: $AIC = 2k - 2\ln(L)$,

where: k is the number of parameters in the model, L is the maximized value of the likelihood function for the model

BIC: Bayes Information Criterion; similar to AIC but using a different dof adjustment (AIC and BIC are only meaningful during comparison of models. The smaller the value, the better the model.)

nParam: Number of parameters (fixed effects parameters+random effects parameters+residual error (eps) parameters)

nObs: Number of observations

nSub: Number of subjects

EpsShrinkage: Epsilon shrinkage, computed as 1-standard deviation (IWRES) for the IWRES values shown on the Residuals worksheet. In cases of multiple dependent variables, the value in the output represents the shrinkage for the last one. Refer to the Core Status tab for the values for each epsilon.

Condition: An estimate of the degree of collinearity in the linearized design matrix. A high condition number may be an indicator of a poor experimental design.

Note: For covariate searches, this is the only output worksheet generated. To generate the full results, change the Run Mode to **Scenarios** in the Run Options tab and re-execute. Since the best scenario is selected automatically after the covariate search, it will be used during the Scenarios run.

PartialDerivatives (Individual; Simple, Simulation): Reports the partial derivative of each prediction for each fixed effect.

Posthoc (Population; Simple, Predictive Check, Simulation): Reports post hoc values. The posthoc parameters are defined in the form of PML in stparm statements, as for example:

$stparm(V=tvV*\exp(nV))$
$stparm(Cl=tvCl*\exp(nCl))$

Here, tvV and $tvCl$ are fixed effects, and nV and nCl are random effects. The numerical values at the posterior mode for each subject (or in the case of QRPEM, the posterior mean) of the empirical Bayesian distribution of nV and nCl are known as the posthoc values of the random effects nV and nCl. The table of posthoc parameters is created from the corresponding stparm formulas with using the optimal value of the fixed effects at the completion of the fit together with the posthoc values of the random effects.

PredCheck_BQLFraction (Population; Predictive Check): Created when a categorical covariate and BQL are involved, and **BQL Fraction** is specified. In addition to the stratum-bins, binning minimum and maximum and IVAR values, the observed and predicted fractions, the name of the observation and the observed quantile levels (Q), are reported.

PredCheck_Cat_ObsQ (Population; Predictive Check): Created when the models involve categorical and/or count observation data. All the actual observations for each categorical/count observed variable are collected at each stratum-bin from the original dataset, and the fraction of observations for each of its observed category or grouped level is calculated.

PredCheck_Cat_SimQ (Population; Predictive Check): Created when the models involve categorical and/or count observation data. The simulated observations for each categorical/count observed variable are collected at each stratum-bin, and the quantile values for the requested quantile levels (column Q) are obtained by summarizing the simulated fractions of observations over all the replicates for each simulated category or grouped level.

PredCheck_ObsQ_SimQCI (Population; Predictive Check): Created when the models involve continuous observed variables. All the actual observations are collected at each stratum-bin from the original dataset, sorted, and the quantiles for the requested percentiles are identified. These are the observed quantiles values summarized at each bin time (Column IVAR) and listed in the DV0 column.

Minimum and maximum values for the bins are listed (Columns Bin_Min and Bin_Max). If confidence intervals for the simulated quantiles were requested, they are summarized (Column DV) and both the level of the confidence interval (Column QE) and the quantile for the confidence intervals (Column QI) are reported. Since each simulated replicate is like the original dataset, quantiles can be calculated in each replicate. Quantiles of each stratum-bin-observed quantile are the secondary confidence intervals. So, for example, the row with QI=50%, QE=10% is the 10% confidence interval of the prediction of the 50% quantile.

PredCheck_SimQ (Population; Predictive Check): Created the models involve continuous observed variables. The requested quantiles are collected for each stratum and bin. In that stratum-bin, all the simulated observations are sorted and the quantiles are calculated and reported. This worksheet lists the quantiles values (Column DV) for the requested quantiles levels (Column Q) summarized over all the replicates and all IDs. Minimum and maximum values for the bins are listed (Columns Bin_Min and Bin_Max). Quantiles are summarized at each bin time (Column IVAR) and if applicable for each stratification level.

PredCheck_TTE (Population; Predictive Check): Created when the **Time-to-event** option is checked on the observation tab. Presents Kaplan-Meyer plot for the chosen dependent variable.

PredCheckAll (Population; Predictive Check, Simulation): For each replicate (Column Replicate) and observation name (Column Obsname), it lists the prediction (Column DV) at each time point (Column IVAR) per individual (usually Column ID5 if only 1 ID column) per sort (Column ID1 to ID4) and per stratum (Column STRAT). The STRAT column contains the actual covariate value. If no stratification is selected the STRAT column will have values of 0 for all rows.

Profile (Population; Profile): Summaries of profile runs indicating the parameter, estimate, log-likelihood, return code, delta (perturbation amount) and percent (perturbation percentage) are reported.

Resid2 (Individual/Population; Simple, Scenarios): When two residual errors are requested, this worksheet reports model predictions and residuals for the second residual error.

Residuals (Individual/Population; Simple, Scenarios): Reports model predictions and residuals. Residuals are "noise" that is not explained by the model.

Secondary (Individual/Population; Simple, Scenarios, Predictive Check, Simulation): Reports secondary parameters specified by a user, including:

– estimated secondary parameters (Columns Parameter and Estimate)
– units (Column Units)
– standard errors (Column Stderr)
– relative standard error (also called coefficient of variation) percentage (Column CV%, calculated as 100* Stderr/ParameterValue)
– confidence limits (upper and lower 95%, 2.5% CI, and 9.5%CI)
– variance inflation factor (Column Var. Inf. Factor)

Simulation (Individual; Simulation): The sort variables, requested IVAR, Simulated DV for the requested Y variables and the name of the Y variables (YVarName) are reported.

Simulation Table 01, **Simulation Table 02**, … (Individual/Population; Predictive Check, Simulation): Created when simulation tables are requested or the **TTE** checkbox is active in **Predictive Check** mode.

StrCovariate (Population; Simple, Scenarios, Predictive Check, Simulation): Table by subject of individual continuous covariates and individual model estimated structural parameter values. This worksheet is used to create scatter plots of covariate versus structural parameters to visualize potential covariate effects.

StrCovariateCat (Population; Simple, Scenarios, Predictive Check, Simulation): Table by subject of individual categorical covariates and individual model estimated structural parameter values. This

worksheet is used to create box-plots of covariate versus structural parameters to visualize potential covariate effects.

Table01 to **Table05** (Optional) (Simple): Optional tables created by the simple run mode or simulation mode for individual models if the user requests them under the Tables Tab. The content depends on what the user enters.

Theta (Individual/Population; Simple, Scenarios, Predictive Check, Simulation): Reports the following:

– estimated fixed effects and standard deviation (eps) parameters (Columns Parameter and Estimate)
– units (Column Units)
– standard errors (Column Stderr)
– relative standard error (or coefficient of variation) percentage (Column CV%), calculated as 100* Stderr/ParameterValue
– confidence limits (upper and lower 95%, Columns 2.5% CI and 9.5%CI)
– variance inflation factor (Column Var. Inf. Factor)

ThetaCorrelation (Individual/Population; Simple, Scenarios, Predictive Check, Simulation): Submatrix of overall parameter correlation matrix corresponding to theta (fixed effect and eps parameters).

ThetaCovariance (Individual/Population; Simple, Scenarios, Predictive Check, Simulation): Submatrix of overall parameter covariance matrix corresponding to theta (fixed effect and eps parameters).

VarCovar (Population; Simple, Scenarios): Variance-Covariance matrix of all the parameter estimators (thetas and omega).

Return Codes

The success of a Phoenix model fit can usually be diagnosed by the engine return code. The return code is a numeric integer value, usually from 1 to 7, but it can also be zero or negative in special cases when convergence is not requested or achieved. Return codes are presented in the Overall Worksheet(s) in the RetCode column, as well as in the Core Output and Core Status text output.

All Phoenix NLME methods, except QRPEM, are based on the direct numerical optimization of a log likelihood or approximate log likelihood function. The fundamental optimization algorithm used is the well-known unconstrained BFGS quasi-Newton UNCMIN optimizer presented as pseudo-code in Dennis, J.E. and Schnabel, R.B., *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, SIAM, 1996 (re-issue of classic 1983 book). Several customized versions of UNCMIN based on this pseudo-code have been implemented in Phoenix for the various engines. The return codes in Phoenix for all engines, except QRPEM, are usually based directly on the ITRMCD return codes of 1 through 5 in UNCMIN. Additional codes of 0, 6, 7, and negative return codes that do not appear in the original UNCMIN have been added to deal with special situations and, in some cases, the meaning of a return code of 4 has been changed. The actual methods of determining the return codes in UNCMIN are rather technical. What follows is a brief summary and the interested user should refer to the reference mentioned above for more detail.

Basically, UNCMIN return codes of 1, 2, and 3 indicate that the optimizer was ‘probably successful’ in reaching at least a local optimum of the log likelihood function, with 1 giving the strongest evidence of success and 3 the weakest. More specifically:

1 = The ‘scaled gradient’ was reduced below a threshold

2 = Further progress does not seem possible due to relative changes in the parameters becoming too small on successive iterations

3 = The line search step in the quasi-Newton direction failed to locate a sufficiently better objective value than the current value

4 = The user-defined maximum number of iterations was reached before any other termination criterion was achieved (note that the meaning of this code has been changed as described below for the FO, Naive pooled, FOCE L-B, and IT2S-EM engines)

5 = The UNCMIN algorithm failed to compute an acceptable step-size within the allowable internal limit on number of trials.

The FO, Naive pooled, FOCE L-B, and IT2S-EM methods in Phoenix require a sequence of UNCMIN optimizations of a log-likelihood or approximate log-likelihood function. Typically, successful convergence is recognized if the final two members of this sequence give essentially identical results with a UNCMIN return code of 1, 2, or 3. Thus, at least two 'successful' UNCMIN optimizations are required for convergence. If convergence is achieved, the return code of 1, 2, or 3 for these engines is the return code from UNCMIN for the final run in the sequence. If these engines do not converge within the user-specified number of UNCMIN attempts (for these engines, the iteration counter is the number of runs of UNCMIN), a return code of 4 is made. If the sequence of UNCMIN optimization is determined to be non-convergent before hitting the maximum number of iterations, a negative return code is made, where the absolute value of the return code is the UNCMIN return code from the final UNCMIN run in the sequence.

Unlike the other UNCMIN-based engines, the FOCE ELS method can sometimes be determined to have converged based on a single UNCMIN run, if the UNCMIN return code is 1 or 2. For a return code of 3, the FOCE ELS algorithm attempts a restart from the final solution so obtained. Here, the iteration counter counts the total number of line searches made throughout all restarts, and thus a return code of 4 means that the total number of such line searches over all attempts has hit the user-specified maximum. If the FOCE ELS algorithm cannot improve upon its original attempt that resulted in a return code of 3, then the results of that attempt are reported with a return code of 3. If, however, better (higher overall log-likelihood) results are obtained, the current attempt then becomes the reference and a new UNCMIN run attempt is made to improve the log-likelihood. The final return code reflects the return code of the best attempt.

To these basic UNCMIN return codes, several others have been added, include 0, 6, 7, and various negative return codes indicating non-convergence. The following summarizes the meaning of the non-negative integer return codes within the context of each method. Negative return codes generally apply in situations where convergence is not achieved and are also described below.

0: The user ran the engine in evaluation model only, no optimization was requested, and the solution that is returned was the same one entered as a starting solution. (Not used in the original UNCMIN algorithm.)

1: Convergence was achieved and the final UNCMIN optimization successfully terminated with a scaled gradient below an internal tolerance (this is the 'best' possible return code). For QRPEM, a return code of 1 indicates successful convergence.

2: Convergence was achieved and optimization of the final UNCMIN run terminated when parameter change fell below internal tolerance on successive iterations of the final UNCMIN run. QRPEM does not use a return code of 2.

3: Convergence was achieved and optimization of the final UNCMIN run terminated when insufficient objective function improvement was made during the line search step along the quasi-Newton direction. QRPEM does not use a return code of 3.

4: For FO, FOCE L-B, Naive pooled, and IT2S-EM, the maximum number of UNCMIN runs was reached before any other termination criterion was achieved. For FOCE ELS, the maximum total number of linear searches over all UNCMIN runs was reached before convergence was achieved. For QRPEM, the maximum number of QRPEM iterations was reached before convergence was achieved, where a QRPEM iteration is one complete cycle through sampling all subjects and making a parameter update based on the samples.

5: Optimization was terminated due to internal failure of UNCMIN to find a suitable step size (this is an error termination).QRPEM does not use a return code of 5.

6: The user manually terminated the engine from the user interface **Stop Early** button in the Progress window before convergence. (Not used in the original UNCMIN algorithm.)

7: The user manually terminated the engine from the user interface **Stop Early** button in the Progress window before convergence. (Not used in the original UNCMIN algorithm.)

Negative return codes generally indicate some kind of convergence failure and often a more definitive reason for the convergence failure can be found in the **Core Status** file in the Text Output section of the Results tab or in the file `nlme7engine.log` (if command line invocation is used). For UNCMIN-based engines, the absolute value of the return code represents the UNCMIN return code on the UNCMIN run that resulted in a determination of convergence failure. QRPEM uses a return code of -4 if non-convergence is detected.

Relatively uncommon, the user can see the return code 40 for FOCE ELS. A code of 40 indicates that the engine went through numerous cycles and was unable to find a solution better than the previous solution by a certain tolerance (.01).

For the Naive Pool engine only, Variance Inflation Factors (VIF) are computed per time entry during simulation runs and presented in several worksheets that tabulate parameter estimates. These are calculated in the same manner as Least-Squares Regression models but are computed regardless of whether the model is simulated or if a fit is run.

VIFs are based on a simple linearization of the model. If the Naive pool engine is run with Number of Iterations > 0, then the linearization is done at the final fitted parameter values. If it is just a simulation, where the model evaluation with Number of iterations=0, then the linearization is done at the initial input values of the parameters. In this case, the observed data values never enter into the VIF computation.

For simulations, VIFs can be used to compare experimental designs. For fits, if the square roots of the VIFs are multiplied by the epsilon estimate it gives standard error estimates based on the linearized model. Standard error of estimates are normally computed from the Hessian of the objective function. If the Hessian is not successful, the Maximum Likelihood Models object will then automatically report the standard errors based on successful VIF computation instead. If this occurs, it is indicated in the Core Status text output.

Plot output

Phoenix model plot output depends on the run mode selected and whether the model is population or individual. Most plots for non-population models are prefixed with “Ind” while population plots are prefixed with “Pop”. Users can always disable and enable plots manually in the Plots tab. The table below describes the content of each potential plot that the Maximum Likelihood Models object creates.

Users can double-click a plot in the Results tab to display it in a separate window for editing. (See the menu options discussion in the Plots chapter of the Data Tools and Plots Guide for plot editing options.)

Double-clicking a point in one of these plots will open the worksheet used to create the plot in a separate window and highlight the corresponding row of data.

Boot Omega Histogram (Population; Bootstrap): Histogram plot of omega elements for all bootstrap replicates.

Boot Theta Histogram (Population; Bootstrap): Histogram plot of fixed effects, residual error, and secondary parameters (if defined) for all bootstrap replicates.

Convergence (Population; Simple, Scenarios): Plots the values for each model parameter at each iteration.

Covariate Box Plots (Population; Simple, Scenarios): Box plots of specified categorical covariates vs Etas. Covariates of interest need to be specified but covariate effects do not need to be part of the model to obtain these plots. ***Note:*** The labels in these plots will contain the values of the categorical covariate, even if a categorical name was specified in the Covar. Type tab.

Covariate Plots (Population; Simple, Scenarios): Scatter plots of specified continuous covariates vs Etas. Covariates of interest need to be specified but covariate effects do not need to be part of the model to obtain these plots. These plots are created for all covariates but, if a covariate changes over time, the first value is selected for plotting purposes, which might make that specific plot meaningless. This will often occur with plots where dose is a covariate.

CWRES Histogram (Population; Simple, Scenarios): Histogram plot of the conditional weighted residual values. If scenarios are present, each histogram is presented by scenario.

CWRES vs IVAR (Population; Simple, Scenarios): Plot of CWRES (conditional weighted residuals; a proposed replacement for the classical WRES (weighted residuals) goodness of fit statistic) against IVAR (the independent variable; typically *time* in a PK fit, *concentration* or *dose* in a PD fit). Values of CWRES should be approximately N(0,1) and hence concentrated between y= –2 and y=+2. Values significantly above 3 or below –3 are suspect and may indicate a lack of fit and/or model misspecification.

CWRES vs PRED (Population; Simple, Scenarios): Same as CWRES vs IVAR, but with the population predictions (i.e., the predictions obtained by setting the random effect values to zero) used for the X-axis.

CWRES vs TAD (Population; Simple, Scenarios): Same as CWRES vs IVAR, but independent variable is Time After Dose.

DV vs IPRED (Individual/Population; Simple, Scenarios): Plot of the dependent variable (DV, e.g., concentrations for PK models) versus individual predicted values (IPRED, e.g., predicted concentrations). Individual prediction obtained by setting random effects to the 'post hoc' or empirical Bayesian estimate of the random effects for the individual from which the DV observation was made. Thus, the plot shows observed vs fitted values of the model function. Ideally, points should fall close to the line of unity y=x.

DV vs IPRED Lattice (Individual; Simple): Same as DV vs IPRED plots for individual models, but latticed by sort.

DV vs IPRED Log (Population; Simple, Scenarios): DV vs IPRED plot for population models, with log transformed DV and IPRED axes.

DV vs PRED (Population; Simple, Scenarios): Analog of DV vs IPRED with population predictions used instead of individual predictions. Since population predictions are typically less accurate, this plot will show larger deviations around the y=x line of unity than DV vs IPRED.

DV vs PRED Log (Population; Simple, Scenarios): DV versus PRED plot with log transformed DV and PRED axes.

DV vs TAD (Population; Simple, Scenarios): Plot of observations vs Time After Dose.

DV, IPRED vs IVAR (Individual/Population; Simple, Scenarios): Plot of the dependent variable (DV), and individual predicted estimates (IPRED) versus the independent variable (IVAR, e.g., time). Ideally, points should fall close to the line of unity y=x.

DV, IPRED vs IVAR Lattice (Individual/Population; Simple, Scenarios): Same plot as DV, IPRED vs IVAR, but latticed by sort.

DV, IPRED vs DV2, IPRED2 Lattice (Individual/Population; Simple, Scenarios): Plots (latticed by individual) of DV (dependent variable), IPRED (individual prediction) vs a second dependent variable (DV2), and a second individual prediction (IPRED2). Generated when there are two residual errors requested.

DV2, IPRED2 vs DV, IPRED Lattice (Individual/Population; Simple, Scenarios): Inverse of DV, IPRED vs DV2, IPRED2 Lattice. Generated when there are two residual errors requested.

DV, IPRED vs TAD (Individual/Population; Simple, Scenarios): Plot of dependent variable and individual prediction vs Time After Dose.

DV, IPRED vs TAD Lattice (Individual/Population; Simple, Scenarios): Same plot as DV, IPRED vs TAD, but latticed by sort.

DV, IPRED, PRED vs IVAR Lattice (Population; Simple, Scenarios): Plots (latticed by individual) containing all population observations, population prediction, and individual predictions vs the independent variable.

DV, IPRED, PRED vs TAD Lattice (Population; Simple, Scenarios): Plots (latticed by individual) containing observations, population prediction, and individual predictions vs Time After Dose.

DV, PRED vs IVAR (Population; Simple, Scenarios): Plots containing observations and population prediction vs the independent variable.

DV, PRED vs IVAR Lattice (Population; Simple, Scenarios): Same plot as DV, PRED vs IVAR, but latticed by individual.

DV, PRED vs TAD (Population; Simple, Scenarios): Plots containing observations and population prediction vs Time After Dose.

DV, PRED vs TAD Lattice (Population; Simple, Scenarios): Same as DV, PRED vs TAD, but latticed.

Eta Histogram (Population): Histogram plot of the eta values. If scenarios are present, each histogram is presented by scenario.

Eta QQ (Population; Simple, Scenarios): Quantile-quantile plot for each eta in the model. If the components of eta are well described by a normal distribution, plotted values will fall roughly along a straight line of unity y=x. Significant deviations from normality, particularly in the tails of the distribution, can be seen by deviations from this line.

Eta Scatter (Population; Simple, Scenarios): Scatter plot of all combinations of etas also known as scatter plot matrix of etas. This allows visual confirmation of correlations between random effects.

IWRES vs IPRED (Individual/Population; Simple, Scenarios): Plot of individual weighted residuals (IWRES) versus individual predicted values (IPRED, e.g., predicted concentrations). Ideally, the blue

line should be at 0 and the red line (with its negative reflection) should not show any fanning. Fanning indicates room for improving the distribution of residuals.

IWRES vs IPRED Lattice (Individual; Simple): Plots (latticed by sort) of individual weighted residuals (IWRES) versus individual predicted values (IPRED, e.g., predicted concentrations).

IWRES vs IVAR (Individual; Simple): Plot of individual weighted residuals (IWRES) versus the independent variable (IVAR, e.g., time). Ideally, the blue line should be at 0 and the red line (with its negative reflection) should not show any fanning. Fanning indicates room for improving the distribution of residuals.

IWRES vs IVAR Lattice (Individual; Simple): Plots (latticed by sort) individual weighted residuals (IWRES) versus the independent variable (IVAR, e.g., time).

IWRES vs TAD (Individual; Simple): Plot of individual weighted residuals (IWRES) versus time after dose.

IWRES vs TAD Lattice (Individual; Simple): Plots (latticed by sort) of individual weighted residuals (IWRES) versus time after dose.

NP Covariate Box Plots (Population; Simple, Scenarios): Box plots of specified covariates vs nonparametric Eta.

NP Covariate Plots (Population; Simple, Scenarios): Scatter plots of specified covariates vs nonparametric Eta.

NP Eta Scatter (Population; Simple, Scenarios): Scatter plot of all combinations of nonparametric Etas.

Partial Derivatives (Individual; Predictive Check): Plot of the partial derivative of each prediction with respect to each fixed effect (which is 1-to-1 with structural parameters).

PCWRES vs IVAR (Population; Simple, Scenarios): Plot of PCWRES (predictive check weighted residuals; a proposed replacement for the classical WRES (weighted residuals) goodness of fit statistic) against IVAR (the independent variable; typically *time* in a PK fit, *concentration* or *dose* in a PD fit). Values of CWRES should be approximately N(0,1) and hence concentrated between y= –2 and y=+2. Values significantly above 3 or below –3 are suspect and may indicate a lack of fit and/or model misspecification.

PCWRES vs PRED (Population; Simple, Scenarios): Same as PCWRES vs IVAR, but with the population predictions (i.e., the predictions obtained by setting the random effect values to zero) used for the X-axis.

PCWRES vs TAD (Population; Simple, Scenarios): Same as PCWRES vs IVAR, but independent variable is Time After Dose.

Posthoc Histogram (Population; Simple): Histogram plot of the posthoc values. If scenarios are present, each histogram is presented by scenario.

PredCheck BQLFraction (Population; Predictive Check): Plot created when **BQL Fraction** mode is specified.

PredCheck ObsQ_SimQ (Population; Predictive Check): Plot created when the model involves continuous observed variables. For each stratification, the observed quantiles are superimposed with the predictive check quantiles over the observed data.

The default color scheme is as follows:
- Red lines: observed quantiles
- Black lines: predicted quantiles
- Blue symbols: observed data

PredCheck ObsQ_SimQCI (Population; Predictive Check): Plot created when the model involves continuous observed variables.

For each stratification, the observed quantiles and the confidence intervals of the simulated quantiles are plotted (with observed data overlaid).

The default color scheme is as follows:
- Black lines: observed quantiles
- Gray lines: confidence intervals of the predicted quantiles
- Blue symbols: observed data

The meaning of the default legend is:
- 05%.05%=5% confidence interval of the 5% quantile of simulated data
- 05%, 95%=95% CI on the 5% quantile of simulated data
- 05%=5% quantile of observed (raw observations)

and so on.

PredCheck Cat ObsQ_SimQ (Population; Predictive Check): A series of plots created when the models involve categorical and/or count observations. The plots show the observed fraction and quantiles of simulated fraction for each category level or grouped level versus the chosen X-axis.

PredCheck_TTE (Population; Predictive Check): Created when the **Time-to-event** option is specified on the observation tab. This is a Kaplan-Meier plot that describes the fraction of patients that reached some state to a specific point of time with confidence interval of observations.

QQ CWRES (Population; Simple, Scenarios): A qq (quantile-quantile) plot of the components of CWRES.
If the components of CWRES are well described by a normal distribution, plotted values will fall roughly along a straight line of unity y=x. Significant deviations from normality, particularly in the tails of the distribution, can be seen by deviations from this line. Can be considered as a diagnostic of model mis-specification.

QQ IWRES (Individual/Population; Simple, Scenarios): Quantile-quantile plot of the individual weighted residuals.
If the components of IWRES are well described by a normal distribution, plotted values will fall roughly along a straight line of unity y=x. Significant deviations from normality, particularly in the tails of the distribution, can be seen by deviations from this line. Can be considered as a diagnostic of model misspecification.

QQ IWRES Lattice (Individual; Simple): Quantile-quantile plot (latticed by sort) of the individual weighted residuals (IWRES). If the components of IWRES are well described by a normal distribution, plotted values will fall roughly along a straight line of unity y=x. Significant deviations from normality, particularly in the tails of the distribution, can be seen by deviations from this line. Can be considered as a diagnostic of model misspecification.

QQ PCWRES (Population; Simple, Scenarios): Same as QQ CWRES but with PCWRES replacing CWRES.

QQ WRES (Population): Same as QQ CWRES but with WRES replacing CWRES.

Simulation (Individual; Simulation): In Simulation run mode, plots (latticed by subject) the simulated and observed dependent variable versus independent variable (IVAR).

Str Covariate Box Plots (Population; Simple, Scenarios): Box plots of the specified categorical covariates vs fixed effects (i.e., model structural parameters). Covariates of interest need to be specified but covariate effects do not need to be part of the model to obtain these plots.

Str Covariate Plots (Population; Simple, Scenarios): Plots of the specified continuous covariates vs fixed effects (i.e., model structural parameters). Covariates of interest need to be specified but covariate effects do not need to be part of the model to obtain these plots.

WRES Histogram (Population; Simple, Scenarios): Histogram plot of the weighted residuals.

WRES vs IVAR (Population; Simple, Scenarios): WRES (weighted residuals) vs IVAR (independent variable) plot.

WRES vs PRED (Population; Simple, Scenarios): WRES (weighted residuals) vs PRED plot.

WRES vs TAD (Population; Simple, Scenarios): Plot of the weighted residuals vs time after dose.

Note: Default plots from library Phoenix models will always generate plots with Time After Dose (TAD) in the X-axis. These plots are generated by default, even if time is not the x-variable in the model (for example pharmacodynamic Emax models) or if dose is an inappropriate concept in the model. In these cases, these plots just duplicate the plots with IVAR in the X-axis.

IWRES is the Individual Weighted RESidual.

$IWRES = (DV - IPRED)/STD$, where $IPRED$ is the individual prediction and STD is the estimate of the standard deviation of the observation.

For example, for an additive residual error model: $DV = IPRED+eps$, the corresponding STD is given by: $STD = stdev(eps)$, where $stdev(eps)$ is the estimated standard deviation for the normal residual error eps. For a multiplicative (proportional) residual error model: $DV = IPRED + IPRED*eps$, the corresponding STD is given by: $STD = \text{abs}(IPRED)*stdev(eps)$.

Assuming the residual error model is correct, $IWRES$ should be $N(0,1)$ random variable.

WRES vs CWRES: WRES (weighted residuals) is a standardized estimate of the components of the population residual vector (observations-population predictions). If a Gaussian observation and random effect model is correct, the components of the population residual vector for a given individual are approximately multivariate normally distributed with a covariance matrix reflecting correlation induced by the fact that all observations from a given individual share a common set of random effect values specific to that individual. The computation of WRES decorrelates these values and standardizes them to unit variance, so the components of WRES are approximately independent N(0,1) random variables. Plots of WRES against population predictions (or QQ plots of WRES) are often used as a diagnostic of model misspecification, excessive deviations from the nominal N(0,1) distribution being regarded as indicative of model misspecification.

Recently, Hooker, Staatz, and Karlsson (2007) noted that WRES is not always a reliable indicator of model misspecification when an FOCE method is used and gave an example where the WRES plot from a highly mis-specified model was better, or closer to what would be expected from the nominal N(0,1) case, than the WRES plot from a correctly specified model. They proposed a modification to WRES as CWRES (“conditional WRES”) which, in the case of correctly specified models, often results in a statistic for which the N(0,1) approximation is better than in the WRES case. In the example, CWRES correctly differentiates between the correctly specified and mis-specified model whereas WRES is completely misleading. CWRES has been gaining acceptance and is one of the diagnostic outputs for POP PK modeling suggested by European Medicines Agency guidelines.

Text output

Compiler Output: The output from compiling the model.cpp file, the C++ language source program that results from building a model in NLME or in Phoenix Modeling Language text. Any errors in compilation will be noted here. If there are any errors, the model fitting process will not run.

Core Output: A text version of output from the running the fitting process. Includes fitted parameter values, eigenvalues, condition number of the matrix of partial derivatives (square root of the ratio of the largest to the smallest eigenvalue), log likelihoods, post hoc estimates, shrinkage values for each epsilon, and a table of goodness-of-fit items (data values, individual and population predictions and residuals, etc.).

Core Status: A text version that lists the minimization process, the minimization engine used, as well as a summary of the Optimal Parameter Estimates and eta shrinkage statistics. (See the Omega worksheet description for information on shrinkage computations.)

Model Text: The PML (Phoenix Modeling Language) code that is generated as the model is specified, plus the column definitions generated when a dataset is mapped.

Remote log: If the model was executed remotely, the log can be reviewed here.

Settings: The settings sent into the compiler and runtime engine. Includes mappings and model code.

Status Window Text: Lists the text that appeared in the Status Window during execution.

Stepwise Text: Stepwise Run Mode only. Explains decision on which covariates to add or subtract in the next step of the covariate search.

Warnings and Errors: If any runtime errors are encountered, they are written to this file.

Additional output

Additional output consists of files that are not be viewed within Phoenix but can be exported and viewed in other software. Additional output file remain part of a Phoenix project but they cannot be used as input to any Phoenix object downstream. Often files created as additional output have the potential to be very large (e.g., simulation). If the user wishes to export and import back these files into Phoenix, caution should be taken to ensure sufficient memory resources.

BootSubj(B).csv: Created for bootstrap mode run. Table showing the subjects that were randomly included in each sample as well as the numbers of tries for those samples.

dmp.txt: These files are created for population models with the Simple mode. The intention is for R users to be able to post-process these files. The dmp.txt files can be loaded into R and they can be used for processing of Phoenix NLME results in R. Individual eta shrinkage statistics are listed at the end of the file. See the Omega worksheet description for information on shrinkage computations.

Rawout.csv: These files list the estimates of Thetas, Omega, and Log-Likelihood for each executed sample.

Rawsimout.csv: These files list the simulation estimates for each executed sample. When running simulation in an individual model, one Rawsimout.csv file is generated for each subject.

Rawsimtbl01.csv–Rawsimtbl05.csv: Optional tables created by the Predictive Check mode, if Tables are requested under the Tables tab and the size is too large to load into Phoenix. The content depends on what the user enters.

Additional Simulation output

Optionally, if a user selects a folder in which to copy simulation results then csv files are saved to that directory. This is only available for population models in Simulation run mode.

Predout.csv: The same as the PredCheckAll worksheet described above (see "Worksheet output").

simtbl01.csv–simtbl05.csv: Optional tables created by the Simulation mode, if **Add Sim Table** is used on the Options tab. The content depends on what the user enters.

ODE error messages

When the ODE solver returns an error code, Phoenix NLME reports the error messages to the user so that appropriate actions may be taken. The error messages may appear in either Core Status text

output or Warnings and Errors text output or both. If any ODE error message appears in the Warnings and Errors text output, then the corresponding results obtained, if there are, should not be trusted and/or be interpreted with care.

For the estimation mode, the ODE error message may appear in either Core Status text output or both Core Status text output and Warnings and Errors text output. If there are occasional occurrences of ODE error messages during some early or intermediate iterations, then it is probably due to some unrealistic parameter values found during the intermediate search, and hence there is typically no need to worry about these (as the optimization is eventually able to go to the right direction). However, if the error message appears during almost all the iterations, then the engine may stuck in a bad/inappropriate region, and hence the estimation results obtained may not be reliable. Moreover, if the error message continues to show up during the standard error calculations step and/or during the final stage for preparing worksheet outputs (e.g., Residual and Overall worksheets), then the corresponding results obtained should not be trusted.

For the simulation modes (including VPC), a 0-iteration fit is performed before the simulation run to populate worksheet outputs. Errors that occur during the 0-iteration fit or simulation runs are recorded in files and reported in the Warnings and Errors text output and the corresponding results obtained, if there are any, should not be trusted. For example, if ODE errors occurred during generation of simulation table worksheets, then simulation tables should be taken with care; if ODE errors occurred during predictive check step, predcheck outputs should be taken with care. It is worth pointing out that 0-iteration fit error files are wiped by subsequent simulation runs. To see them, the user should re-run the model in simple mode.

Typically, the ODE errors can be avoided by taking appropriate actions. For example, if the error message is about maximum number of function evaluations exceeded, then one can increase the value of “ODE max step” (by clicking the “Advanced <<” button in the Run Option tab) to be sufficiently large to avoid such error. While, for some cases, the easiest/best way is just to switch to a different solver. For example, if the model is suspected to be stiff, then one needs to switch to auto-detect or stiff solver (from the “max ODE” menu located in the Run Options tab).

Phoenix NLME computations

Differential equations in NLME

Ordinary differential equations can be solved for Phoenix models by several methods. These methods available under the option **Max ODE** in the Run Options tab.

The ODE solvers used by Phoenix have three accuracy controls that are available in the Advanced Run Options tab:

ODE Rel. Tol.: Relative tolerance (RTOL)
ODE Abs. Tol.: Absolute tolerance (ATOL)
ODE max step: Maximum number of allowable steps or function evaluations (depending on the solver) to achieve the above accuracies (MAXSTEP)

For more information, see “RTOL, ATOL, and MAXSTEP ODE error controls”.

Matrix Exponent option

This is the default option for solving ordinary differential equations for Phoenix models. It requires that the system be expressible as $dy/dt=Jy$, where J is a matrix (the Jacobian) with elements that are constant over a time interval.

The general N-compartment model governed by ordinary differential equations takes the form:

$$\dot{y} = f(y) + r$$

where $y = (y_1, \ldots, y_N)^T$ is an N-dimensional column vector of amounts in each compartment as a function of time, $f(y)$ is a column vector-valued function that gives the structural dependence of the time derivatives of y on the compartment amounts, and r is an N-dimensional column vector of infusion rates into the compartments. Note that as opposed to the closed-form solution of first-order models, dosing can be made into any combination of compartments, and all of the compartments are modeled.

To account for infusion rates, define the augmented system:

$$Y = \begin{bmatrix} y \\ 1 \end{bmatrix} \text{ and } R = \begin{bmatrix} r \\ 0 \end{bmatrix}$$

In the special first-order case, the equations are represented as:

$$\dot{Y} = JY$$

where J is the Jacobian matrix given by:

$$J = \begin{bmatrix} \frac{\partial f}{\partial y} & r \\ 0 & 0 \end{bmatrix}$$

The partial derivatives are obtained by symbolic differentiation of $f(y)$. If any of them are not constant (over the given time interval), matrix exponent cannot be used. In such cases, Phoenix automatically switches the ODE solver from the matrix exponent to the DVERK except in the case where the model involves `gammaDelay` function (for this case, it automatically switches to DOPRI5). It is worth pointing out that the actual ODE solver requested/used can be found at the beginning of the Core Output tab (Text Output Results section).

Assuming J is constant, the state vector Y evolves according to:

$$Y(t) = e^{Jt}Y(0)$$

where e^{Jt} is defined by the Taylor series expansion of the standard exponential function. Standard math library routines are available for the computation of the matrix exponential. These are faster and more accurate than the equivalent computation with a numerical ODE solver and do not suffer from stiffness.

In the steady state case where a vector bolus dose is given at intervals of length, the solution takes the form:

$$Y(t) = e^{Jt}(I - e^{J\tau})^{-1}D$$

where D is a column vector giving the amount of the bolus deposited in each compartment.

In the case where the matrix exponent cannot be used, the steady state is found by pre-running the model via the ODE solver for enough multiples of τ so that Y changes by less than a tolerance.

Stiff option — LSODE

LSODE (Livermore Solver for Ordinary Differential Equations) is the ODE solver used by Phoenix to solve stiff systems of the form $dy/dt=f$. It treats the Jacobian matrix df/dy as either a dense (full) or a banded matrix, and as either supplied automatically by Phoenix or internally approximated by difference quotients (if Phoenix fails to generate it).

Stiff systems typically arise from models with processes that proceed at very different rates, for example, very slow equilibration of a peripheral compartment with a central compartment relative to the rate at which elimination occurs from the central compartment. Generally, a stiff solver outperforms, in terms of both time and accuracy, a non-stiff solver for such systems, but the non-stiff solver can be superior for non-stiff systems.

Auto-Detect option — LSODA

Nonlinear systems can change from stiff to non-stiff during evaluation, which can result in incorrect estimates and/or much longer execution times if the incorrect ODE solver is selected. LSODA (written by Linda R. Petzold and Alan C. Hindmarsh) is a variant version of the LSODE solver and it automatically selects between non-stiff (Adams) and stiff (Backward Differentiation Formula) methods. LSODA is a very robust adaptive stepwise solver that uses the non-stiff method initially, and dynamically monitors data in order to decide which method to use.

Non-Stiff option — DVERK

This is an explicit Runge-Kutta method based on Verner's fifth and sixth order pair of formula. Due to its fast and robust performance, it is one of the widely used methods for solving non-stiff problems and is also the default solver used by Phoenix when the matrix exponent solver is not applicable.

It is well-known that the explicit Runge-Kutta methods cannot compete with specifically designed stiff methods, except for very mildly stiff systems. For the stiff problems, integration errors are expected. The user is advised to switch to the LSODE/LSODA solver.

Non-Stiff option — DOPRI5

DOPRI5 is another widely used solver for the non-stiff problems. It is an explicit Runge-Kutta method of order (4)5, based on Dormand-Prince pair. More specifically, it uses six function evaluations to calculate fourth- and fifth-order accurate solutions. Numerical experiments show that DOPRI5 may be more reliable and efficient than the DVERK, in certain stiff problems.

RTOL, ATOL, and MAXSTEP ODE error controls

The default values for ATOL and RTOL are 0.000001, while the default value of MAXSTEP is 50000.

- Matrix Exponent solver
  For the matrix exponent solver, only an absolute tolerance (ATOL) applies. If ATOL is set to zero, it is reset internally to the square root of the machine precision (approximately 1.e-8). For a full discussion, see Sidje, Roger B., "EXPOKIT – A Matrix Package for Computing Matrix Exponentials," *ACM Transactions on Mathematical Software*, 24 (1998) pp. 130–156.
- LSODA/LSODE solver
  The estimated local error for each component, $Y(i)$, is controlled so as to be roughly less (in magnitude) than $RTOL*abs(Y(i)) + ATOL$. Thus, the local error test passes is, in each component, either the absolute error is less than ATOL or the relative error is less than RTOL. If RTOL is set to 0.0, then only ATOL is applied by the solver (pure absolute error control). Currently, the use of pure relative error control (that is, ATOL is set to 0.0) is not permitted for the LSODE/LSODA solver and can lead to errors. Note that both tolerances are local controls on the current integration interval, and the global error may exceed these values.

  For LSODE and LSODA, MAXSTEP is the maximum number of steps allowed during one call to the solver and 'MAXSTEP = 0' means that, at most, 5,000 steps will be used.

  For a more complete discussion of error controls, see Hindmarsh, Alan C., "ODEPACK, A Systematized Collection of ODE Solvers," *Scientific Computing*, R. S. Stepleman et al (eds.), North-Holland, Amsterdam (1983) pp. 55–64.
- DVERK (non-stiff) solver
  For the DVERK solver, only relative tolerance (RTOL) applies and serves as a global tolerance term (that is, the ATOL value is ignored): the solver attempts to control a norm of the local error in such a way that the global error is proportional to RTOL. It is worth pointing out that a global tolerance cannot be set to zero.

  For this solver, MAXSTEP represents the maximum number of function evaluations allowed during one call to the solver, 'MAXSTEP = 0' means that there is no limit in the number of function evaluations.

  For a full discussion, see http://www.cs.toronto.edu/NA/Users.Guide.For.DVERK.pdf.
- DOPRI5 (non-stiff) solver
  The code keeps the local error of $Y(i)$ below $RTOL*abs(Y(i)) + ATOL$. Thus, the local error test

passes if, in each component, either the absolute error is less than ATOL or the relative error is less than RTOL. If RTOL is set to 0.0, then only ATOL is applied by the solver (pure absolute error control). Similarly, if ATOL is set to 0.0, only RTOL is applied by the solver.

For DOPRI5, MAXSTEP denotes the maximum number of steps allowed during one call to the solver and 'MAXSTEP = 0' means that, at most, 100,000 steps will be used.

For a full discussion, see Hairer, Norsett, and Wanner (1993): *Solving Ordinary Differential Equations. Nonstiff Problems*. 2nd edition. Springer Series in Comput. Math., vol. 8.

Prediction and Prediction Variance Corrections

Including a prediction correction and prediction variance correction as part of predictive checking allows observations that would otherwise be incomparable to be pooled together to narrow the quantiles of predicted data, making a more stringent test for possible model misspecification. For example, by using time-after-dose (TAD) as the X axis, data following multiple dose events can be combined. If doses are given at widely varying dose amounts, predictions of plasma concentrations will be proportionally scaled together (assuming a linear model). Similarly, variability correction may apply if, for example, the model is linear but the error model is additive.

Prediction and prediction variance corrections deal with these quantities:

Y_{ij}: j^{th} observation for i^{th} subject
$PRED_{ij}$: Population (zero-eta) prediction of j^{th} observation for i^{th} subject
$PRED_{bin}$: Median of $PRED_{ij}$ over all observations in a particular bin
pcY_{ij}: Prediction-corrected version of j^{th} observation for i^{th} subject

The prediction-corrected observation is calculated either by the proportional rule (default):

$$pcY_{ij} = Y_{ij}\frac{PRED_{bin}}{PRED_{ij}}$$

or by the additive rule:

$$pcY_{ij} = Y_{ij} + (PRED_{bin} - PRED_{ij})$$

Further, if **Pred. Variance** is selected, these variables come into play:

$sd(pcY_{ij})$: Standard deviation of pcY_{ij}
$sd(pcY_{bin})$: Median of $sd(pcY_{ij})$ over the bin
$pvcY_{ij}$: Prediction-variability corrected version of Y_{ij}

$pvcY_{ij}$ is calculated as follows:

$$pvcY_{ij} = PRED_{bin} + (pcY_{ij} - PRED_{bin})\frac{sd(pcY_{bin})}{sd(pcY_{ij})}$$

Then either the quantity $pvcY_{ij}$ or pcY_{ij} is used in the predictive-check plots, depending on whether **Pred. Variance** is selected.

Shrinkage calculation

The Omega output worksheet contains η shrinkage data. It is based on the standard deviation definition:

$$\text{Shrinkage\_SD}_j = 1 - \frac{\text{SD}(\eta_j)}{\sqrt{\omega_{j,j}}}$$

where $SD(\eta_j)$ is the empirical standard deviation of the j[th] η over all $Nsub$ subjects, and $\omega_{j,j}$ is the estimate of the population variance of the j[th] random effect, j = 1,2,…, $NEta$.

For all population engines other than QRPEM, the numerical η_j value used in the shrinkage computation is the ***mode*** (maximum) of the empirical Bayesian posterior distribution of the random effects η_j, evaluated at the final parameter estimates of the fixed and random effects. For QRPEM, the η_j value is the ***mean*** of the empirical Bayesian distribution.

It is worth pointing out that another common way to define the η-shrinkage is through the variance, and is given by

$$\text{Shrinkage\_Var}_j = 1 - \frac{\text{Var}(\eta_j)}{\omega_{j,j}}$$

where $\text{Var}(\eta_j)$ is the empirical variance of the j[th] η over all $Nsub$ subjects. By the two previous equations, one can see that standard deviation based η-shrinkage can be computed from variance based η-shrinkage, and vice versa. For example, if one has $Shrinkage\_Var_j$, then $Shrinkage\_SD_j$ can be calculated as:

$$\text{Shrinkage\_SD}_j = 1 - \sqrt{1 - \text{Shrinkage\_Var}_j}$$

The Eta output worksheet contains the individual shrinkage, which is calculated as follows:

$$\text{Shrink\_Sub\_Var}_{i,j} = \frac{(\eta\_\text{SE}_{i,j})^2}{\omega_{j,j}}$$

where i = 1, 2, …, $Nsub$ and j = 1, 2, …, $NEta$. Here, $\eta\_SE_{i,j}$ denotes the standard error of the j[th] individual parameter estimator for the i[th] subject. For all population engines other than QRPEM, $\eta\_SE_{i,j}$ is calculated as the square root of the (j, j)[th] element of the inverse of the negative Hessian (second derivative matrix) of the empirical Bayesian posterior distribution for the i[th] subject. While, for the QRPEM engine, they are computed via the importance sampling of the empirical Bayesian posterior distribution.

The formula used to calculate $Shrink\_Sub\_Var$ (the previous equation) is extended from the 1-1 relation between the population shrinkage and standard error of individual parameter estimator conjectured in Xu, et al, *AAPS J.*, (2012) pp. 927-936. This relationship can be intuitively observed from the following important theoretical relationship obtained for the EM algorithm.

$$\omega_{j,j} = \sum_{i=1}^{Nsub} \frac{(\eta\_SE_{i,j})^2}{Nsub} + Var(\eta_j)$$

From which, one can see that the commonly used variance based population shrinkage becomes

$$\text{Shrinkage\_Var}_j = \frac{1}{Nsub} \sum_{i=1}^{Nsub} \frac{(\eta\_SE_{i,j})^2}{\omega_{j,j}}$$

It is worth pointing out that population shrinkage calculated using the above formula is also reported in `bluptable.dat` (after the individual shrinkages for each η), and is denoted as *shrinkageebd_var* (see the highlighted text in yellow in the image below). From these results, we can see that the value of *shrinkageebd_var* is similar to the one for *Shrinkage_Var*.

```
58  1   -0.210141    0.106809    0.446799    0.239053    0.057146    0.02899
59  1   -0.393856    0.097149    0.446799    0.217434    0.047277    0.02392
 -0.01432947  !eta_mean
  0.43313437  !eta_std_dev
  0.05638929  !eta_mean_std_err
  0.25411688  !|etamean|/eta_mean_std_err
  0.03058260  !shrinkage_sdv = 1 - eta_sdv/omegasd(j,j)
  0.06022990  !shrinkage_var = 1 - eta_var/omega(j,j)
  0.06972999  !shrinkageebd_var = etaebd_var/omega(j,j)
```

In `bluptable.dat`, *Shrink_Sub_SD* is calculated by using the following formula

$$\text{Shrink\_Sub\_SD} = 1 - \sqrt{1 - \text{Shrink\_Sub\_Var}}$$

This relationship between *Shrink_Sub_SD* and *Shrink_Sub_Var* is an analog of the relationship shown in the earlier Shrinkage_SD equation.

Maximum Likelihood Models examples

Completed projects for most of these examples are available for reference in …`\Examples\NLME`. You can save a copy of the `Examples` directory (installed with Phoenix) to your Phoenix project directory via the Project Settings in the *Phoenix Preferences* dialog.

Within this directory there are other projects that illustrate one or more aspects of modeling with the Phoenix NLME engines:

- `QRPEM_with_time_varying_covariate.phxproj` shows how to structure a QRPEM model with time-varying covariates so that QRPEM can run it.
- `Remifen.phxproj` uses a more complex well-known model that uses nonlinear covariates (“Influence of Age and Gender on the Pharmacokinetics and Pharmacodynamics of Remifentanil. I. Model Development.” Minto, Charles F.; Schnider, Thomas W.; Egan, Talmage D.; Youngs, Elizabeth; Lemmens, Harry J.; Gambus, Pedro L.; Billard, Valene; Hoke, John F.; Moore, Katherine H.; Hermann, David J.; Muir, Keith T.; Mandema, Jaap W.; Shafer, Steven L. *Anesthesiology*, January 1997, 86(1): 10–23. Clinical Investigation). This model is restructured so that it can be solved with QRPEM.

Most Phoenix objects require the same basic steps for their use. However, there may be multiple paths to accomplishing a step (e.g., main menu, right-click menu, drag-and-drop, etc.).

Variations in models using Theophylline data

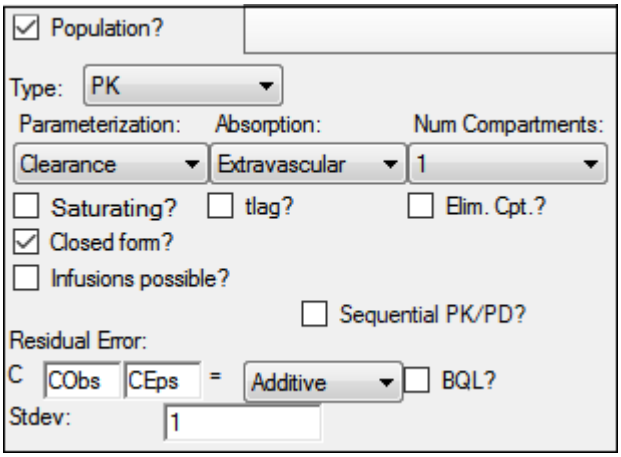
The data in `theopp.dat` came from a study of the kinetics of the anti-asthmatic agent theophylline reported by Boeckmann et al. (1992). In this experiment, the drug was administered orally to twelve subjects, and serum concentrations were measured at ten time points per subject over the subsequent 25 hours.

The common model for the kinetics of theophylline following oral administration is a one-compartment model with first-order absorption and elimination. In this model, the subject-specific pharmacokinetic parameters to be estimated are the absorption rate constant, **Ka**, the volume of distribution, **V**, and the clearance rate constant, **Cl**.

Note: The completed project (`Theo_Model.phxproj`) is available for reference in …`\Examples\NLME`.

Set up the Maximum Likelihood Models object

1. Create a new project called `Theo_Model`.
2. Import the dataset …`\Examples\NLME\Supporting files\theopp.dat`.
   Click **Finish** in the *File Import Wizard* dialog.
3. Right-click on **Workflow** in the Object Browser and select **New > Modeling > Maximum Likelihood Models**.
4. Rename the Maximum Likelihood Models object `Diagonal Model`.
   This model is a one-compartment model with 1st order input and 1st order elimination.
5. Leave **Clearance** in the Structure tab **Parameterization** menu.
6. In the **Absorption** menu, select **Extravascular**.
7. Leave the **Num Compartments** menu set to **1**, leave the **Closed form?** box selected, and leave the error model set to **Additive**.



8. Select the **Parameters** tab and then select the **Fixed Effects** sub-tab.
9. In the **Initial** column, type the following initial estimates for each of the study parameters:
   tvKa = 2
   tvV = 0.5
   tvCl = 0.1
10. Select the **Structural** sub-tab.

11. Click **Add Covariate**.
12. Type `wt` in the **Covariate** field. wt is added as a context association in the Main Mappings panel.

Note: Adding weight as a covariate allows the Maximum Likelihood Models object to create covariate plots in the output.

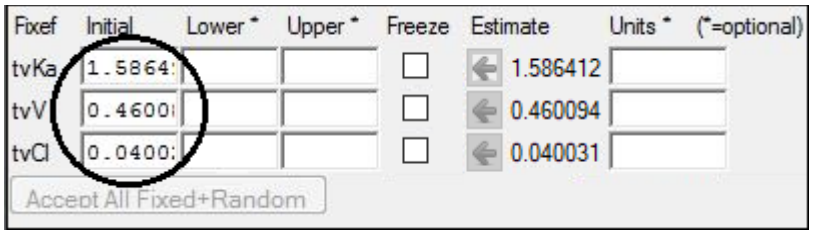
13. Select the **Run Options** tab.
14. Click **Advanced**.
15. In the **SE Step** field that is added to the panel, change the value from `0.01` to `0.001`.

Map the dataset

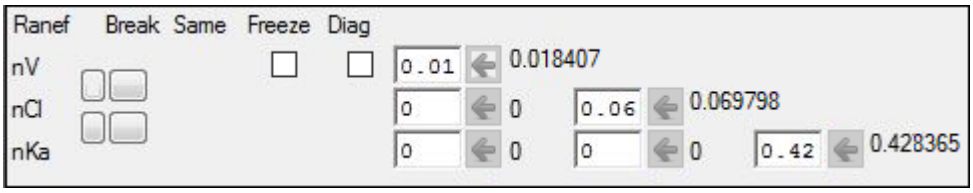
1. Use the mouse pointer to drag the **theopp** worksheet from the Data folder to the Main Mappings panel.
2. Map the data types to the following contexts:
   Map **xid** to the **ID** context.
   Map **dose** to the **Aa** context.
   Map **time** to the **Time** context.
   Map **yobs** to the **CObs** context.
   Map **wt** to the **wt** context.
3. Click  (**Execute** icon) to execute the object.

Set up the full block model

1. In the Object Browser, right-click **Diagonal Model** object and select **Copy**.
2. Right-click the **Workflow** object and select **Paste**.
   The model object and its settings are pasted into the Theo_Model project and named "Copy of Diagonal Model".
3. Rename the project as to `Full Block Model`.
4. Go to the **Parameters > Fixed Effects** sub-tab.
5. Click **Accept All Fixed+Random** to copy the estimates to the Initial estimates field for each parameter.



6. Select the **Random Effects** sub-tab.
7. Clear the **Diag** checkbox.to create a full block model.



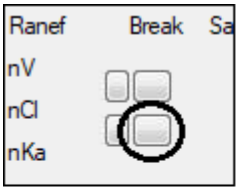
The data types are automatically mapped as follows:

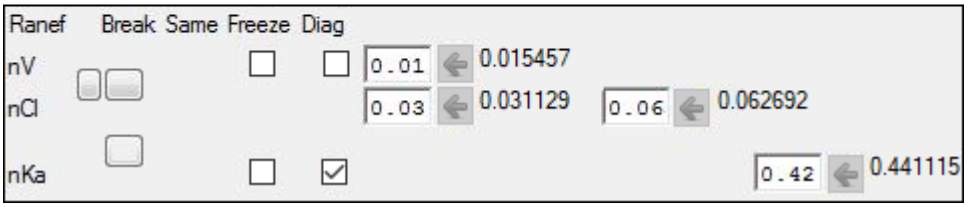
xid to the **ID** context
dose to the **Aa** context
time to the **Time** context
yobs to the **CObs** context
wt to the **wt** context

8. Execute the object.

Set up the partial diagonal model

1. Right-click **Full Block Model** and select **Copy**.
2. Right-click the **Workflow** object and select **Paste**.
   The model object and its settings are pasted into the Theo_Model project and named “Copy of Full Block Model”.
3. Rename the project as `Partial Diagonal Model`.
4. Go to the **Parameters > Fixed Effects** sub-tab.
5. Click **Accept All Fixed+Random** to copy the new estimates to the Initial estimates field for each parameter.
6. Select the **Random Effects** sub-tab.
7. Split the random effect blocks between the parameters Cl and Ka by clicking the **Split or join the random effect blocks** button (under the **Break** heading) between **Cl** and **Ka**.



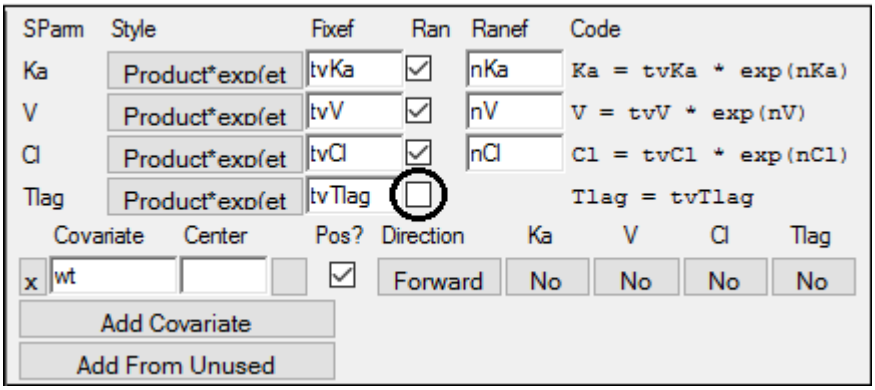


The data types are automatically mapped as follows:
xid to the **ID** context
dose to the **Aa** context
time to the **Time** context
yobs to the **CObs** context
wt to the **wt** context

8. Execute the object.

Set up the partial diagonal model with time lag

1. Right-click **Partial Diagonal Model** and select **Copy**.
2. Right-click the **Workflow** object and select **Paste**.
   The model object and its settings are pasted into the Theo_Model project and named "Copy of Partial Diagonal Model".
3. Rename the project as `Partial Diagonal Model tlag`.
4. Go to the **Structure** tab and check the **tlag?** checkbox to add the Tlag (time lag) parameter to the model.
5. Select the **Parameters > Fixed Effects** sub-tab.
6. Click **Accept All Fixed+Random** to copy the new estimates to the Initial estimates field for each parameter.
7. Select the **Structural** sub-tab.
8. Clear the **tvTlag** checkbox to remove random effects from Tlag.



9. Execute the object.

Save and close the project

1. Select **File > Save Project**.
2. Click **Save**.
3. Select **File > Close Project**.

The project is saved and closed and Phoenix can be safely closed.

This concludes the model variations example.

Variations in models using Phenobarbital data

The data in `pheno.dat` come from a study of the neonatal pharmacokinetics of phenobarbital reported by Grasela and Donn. (Grasela and Donn (1985). Neonatal Population Pharmacokinetics of Phenobarbital Derived from Routine Clinical Data. *Dev Pharmacol Ther,* 8:372–383.).

Data were collected on 59 pre-term infants given phenobarbital for prevention of seizures during the first 16 days after birth. Each individual received an initial dose followed by one or more sustaining doses by intravenous administration. A total of between one and six concentration measurements were obtained from each individual at times other than dose times as part of routine monitoring, for a total of 155 measurements. Additionally, birth weight (wt) and a five-minute Apgar score (Apgar < 5) were recorded for each subject.

The pharmacokinetics of phenobarbital can be described by a one-compartment model with intravenous administration and first-order elimination, which is found in Phoenix library models as PK model 1. In this model, the subject-specific pharmacokinetic parameters to be estimated are the elimination rate constant, Ke, and volume of distribution, V.

Note: The completed project (`Pheno.phxproj`) is available for reference in …`\Examples\NLME`.

Set up the Maximum Likelihood Models object

1. Create a new project called `Pheno`.
2. Import the dataset …`\Examples\NLME\Supporting files\pheno.dat`.
3. Click **Finish** in the *File Import Wizard* dialog.
4. Right-click on **Workflow** in the Object Browser and select **New > Modeling > Maximum Likelihood Models**.
5. Rename the Maximum Likelihood Models object `Pheno Model`.
6. Select the **Parameters > Structural** sub-tab.
7. Click **Add Covariate**.
8. Type `wt` in the **Covariate** field. wt is added as a context association in the Main Mappings panel.
9. Click **Add Covariate**.
10. Type `apgr` in the **Covariate** field. apgr is added as a context association in the Main Mappings panel.

Note: Adding weight (wt) and Apgar (apgr) as covariates allows Phoenix to create covariate plots in the output.

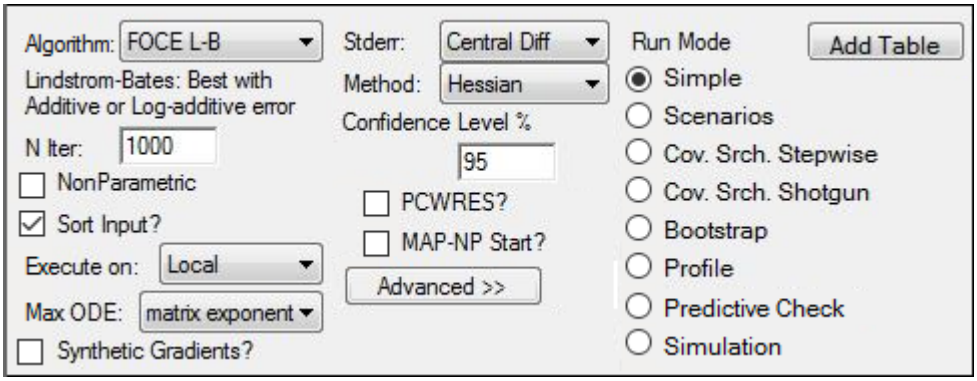
Map the pheno dataset

1. Use the mouse pointer to drag the **pheno** worksheet from the Data folder to the Main Mappings panel.
2. Select the option buttons in the Main Mappings panel to map the data types to the following contexts:
   xid to the **ID** context.
   time to the **Time** context.
   dose to the **A1** context.
   wt to the **wt** context.

apgr to the **apgr** context.
yobs to the **CObs** context.

Leave **idum1** mapped to **None**.
Leave **idum2** mapped to **None**.

3. In the **Structure** tab, select **Micro** from the **Parameterization** menu.
4. Select the **Run Options** tab.
5. Select **FOCE L-B** as the **Algorithm** and **Hessian** as the **Method**.



6. Click (**Execute** icon) to execute the object.
7. In the Results tab, select the **Core Output** text results.

Note that, in the residual error model, the parameter value for fixed effects parameters is used in the next model.

Set up the full block model with standard deviation

1. Right-click **Pheno Model** and select **Copy**.
2. Right-click the **Workflow** object and select **Paste**.
3. Rename the project as `Pheno Stdev`.

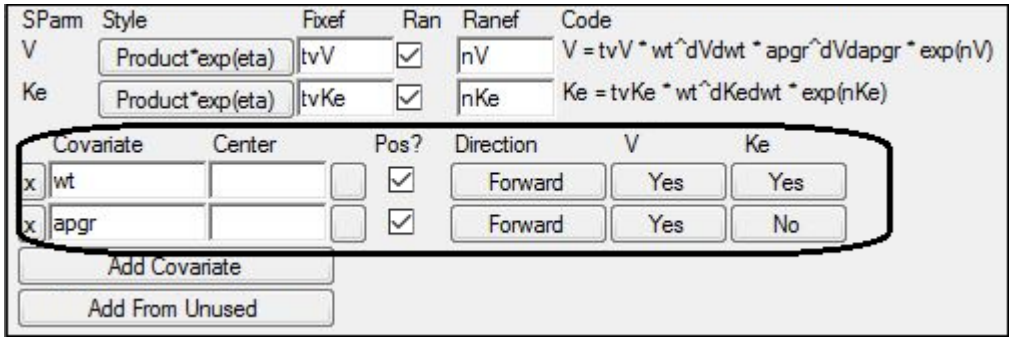
The data types are automatically mapped as follows:
xid to the **ID** context
time to the **Time** context
dose to the **A1** context
wt to **wt**
apgr to **apgr**
yobs to the **CObs** context
idum1 mapped to **None**
idum2 mapped to **None**

4. In the Structure tab, type `2.76` in the **Stdev** field.
5. Select the **Parameters > Fixed Effects** sub-tab.
6. Click **Accept All Fixed+Random** to copy the new estimates to the Initial estimates field for each parameter.
7. Select the **Random Effects** sub-tab.
8. Clear the **Diag** checkbox to use the full block structure.

9. Execute the object.

Add covariate effects to study parameters

1. Right-click **Pheno Stdev** and select **Copy**.
2. Right-click the **Workflow** object and select **Paste**.
3. Rename the project as `Pheno Stdev Covar`.
4. Select the **Parameters > Fixed Effects** sub-tab.
5. Click **Accept All Fixed+Random** to copy the new estimates to the Initial estimates field for each parameter.
6. Select the **Structural** sub-tab.
7. Underneath the **V** parameter, click **No** for the **wt** covariate to change it to **Yes**.
8. Underneath the **Ke** parameter, click **No** for the **wt** covariate to change it to **Yes**.
   The study variable weight is added as a covariate effect to both V and Ke.
9. Underneath the **V** parameter, click **No** for the **apgr** covariate to change it to **Yes**.
   The study variable apgr is added as a covariate effect to V.



The data types are automatically mapped as follows:
xid to the **ID** context
time to the **Time** context
dose to the **A1** context
wt to the **wt** context
apgr to the **apgr** context
yobs to the **CObs** context
idum1 mapped to **None**
idum2 mapped to **None**

10. Execute the object.

Save and close the project

1. Select **File > Save Project**.
2. Click **Save**.
3. Select **File > Close Project**.

The project is saved and closed and Phoenix can be safely closed.

This concludes the model variations example that uses Phenobarbital data.

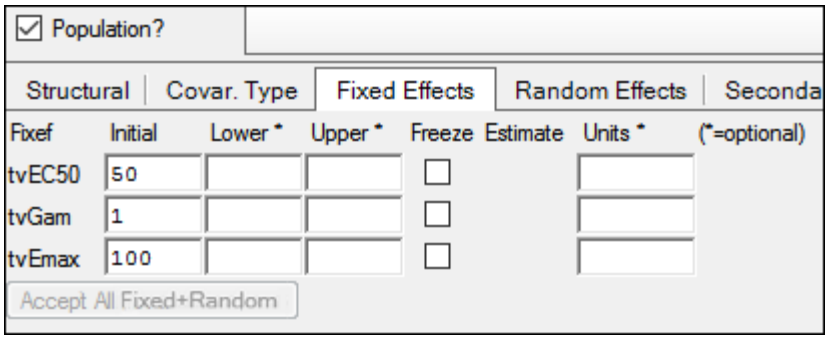
Variations in Emax models

The data in `emax.dat` fits a sigmoid Emax model to dose and effect data from 100 subjects. There are five effect observations for each subject, at the same five dose levels. The model sets effect E equal to a function of the drug concentration C, with a model parameters describing the concentration at half-maximal effect EC50, maximum possible effect Emax, and parameter gamma.

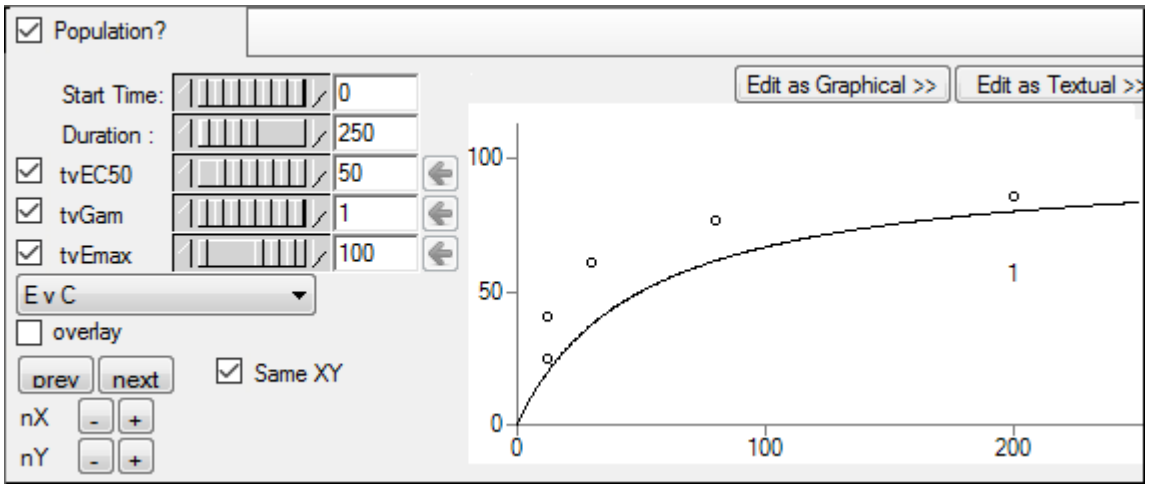
Note: The completed project (`Emax.phxproj`) is available for reference in …`\Examples\NLME`.

Set up Emax model

1. Create a new project called `Emax`.
2. Import the dataset …`\Examples\NLME\Supporting files\emax.dat` and `simhill4.5.dat`.
3. Click **Next Arrow** button to move from emax to the simhill4.5 dataset and click **Finish** in the *File Import Wizard* dialog.
4. Right-click on **Workflow** in the Object Browser and select **New > Modeling > Maximum Likelihood Models**.
5. Rename the object `Emax`.
6. Use the mouse pointer to drag the **emax** worksheet from the Data folder to the Main Mappings panel.
7. In the Structure tab, select **Emax** in the **Type** menu.
8. Check the **Sigmoid** checkbox to specify the model is a sigmoidal emax model.
9. Select the option buttons in the Main Mappings panel to map the data types as follows:
   subject to the **ID** context.
   dose to the **C** context.
   e to the **EObs** context.
10. Select the **Parameters > Fixed Effects** sub-tab.
11. In the **Initial** field for **tvEC50**, enter `50` as the initial estimate value.
12. Leave the initial estimate for **tvGam** set to **1**.
13. In the **Initial** field for **tvEmax**, enter `100` as the initial estimate value.



14. Select the **Initial Estimates** tab to find the best initial estimates.
15. Adjust the portion of the plot to view by setting the **Duration** to `250` using the slider or typing the value in the field next to the slider.

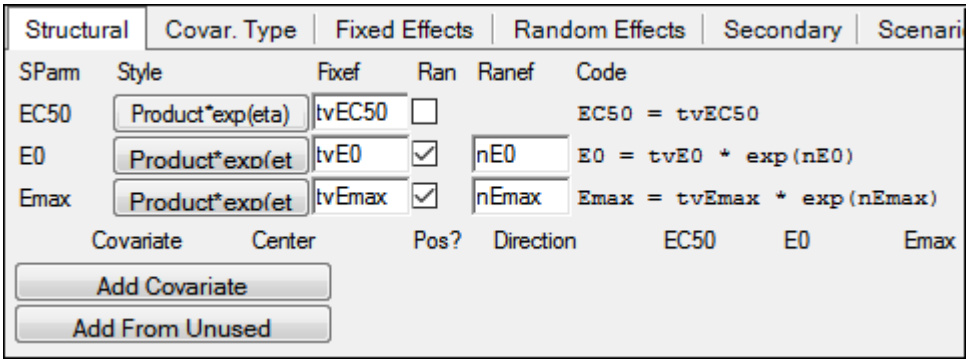


Enter different values for fixed effect parameters, and see the results on the XY plot. To select a negative value with the slider, clear the checkbox beside the parameter name. For more information on using the Initial Estimates, see the "Initial Estimates tab" description.

16. Click (**Execute** icon) to execute the object.

Set up an Emax Baseline project

1. Insert a second Maximum Likelihood Models object into the workflow and name it `Emax Baseline`.
2. In the **Structure** tab, select **Emax** in the **Type** menu.
3. Check the **Baseline** checkbox.
4. Select the **Parameters > Structural** sub-tab.
5. Clear the **Ran** checkbox beside **EC50** to remove the random effect from the parameter.

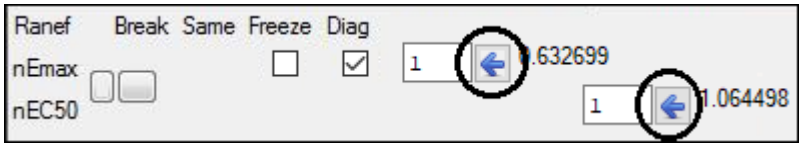


6. Select the **Fixed Effects** sub-tab.
7. In the **Initial** field for **tvEC50**, enter `15` as the initial estimate value.
8. Leave the **Initial** field for **tvE0** set to **1** as the initial estimate value.
9. In the **Initial** field for **tvEmax**, enter `150` as the initial estimate value.
10. Use the mouse pointer to drag the **simhill4.5** worksheet from the Data folder to the Main Mappings panel.

11. Select the option buttons in the Main Mappings panel to map the data types to the following contexts:
ID to the **ID** context.
CONC to the **C** context.
DV to the **EObs** context.
12. Execute the object.

Set up an Emax Baseline project including standard deviations

1. Right-click **Emax Baseline** and select **Copy**.
2. Right-click the **Workflow** object and select **Paste**.
3. Rename the project as `Emax Baseline Stdev`.
The data types are automatically mapped as follows:
ID to the **ID** context
CONC to the **C** context
DV to the **EObs** context
4. In the **Structure** tab, type `50` in the **Stdev** field.
5. Select the **Parameters > Structural** sub-tab.
6. Check the **Ran** checkbox for **EC50**.
7. Clear the **Ran** checkbox for **E0**.
8. Select the **Fixed Effects** sub-tab.
9. Click left arrow button beside **tvEC50** to copy the new estimate to the Initial estimate field for tvEC50.
10. Check the checkbox beside **tvE0** to freeze the fixed effect for the parameter.
11. Click the left arrow button beside **tvEmax** to copy the new estimate to the Initial estimate field for tvEmax.
12. Select the **Random Effects** sub-tab.
13. Click the **Copy omega estimate** left arrow buttons beside **nEmax** and **nEC50** to copy the omega estimates to the initial estimate field for each parameter.



14. Execute the object.

Save and close the project

1. Select **File > Save Project**.
2. Click **Save**.
3. Select **File > Close Project**.

The project is saved and closed and Phoenix can be safely closed.

This concludes the Emax model variations example.

Fitting PK and PD data and using a simulation run

This example demonstrates a simultaneous fitting of PK and PD data. The PK data follows a one-compartment model with first-order absorption and elimination through oral administration. The PD data can be described using a simple Emax model. Both PK and PD models are found in the Phoenix library models as PK model 3 and PD model 102.

This example also demonstrates the use of the Monte Carlo population simulation that is available when users select the Predictive Check run mode option. Simulation runs can be used when modelers have PK/PD parameter estimates from prior modeling runs and want to explore the effects of modifying certain conditions, such as dosing regimens, without having to import or create more data.

Note: The completed project (`PKPD_Linked.phxproj`) is available for reference in …`\Examples\NLME`.

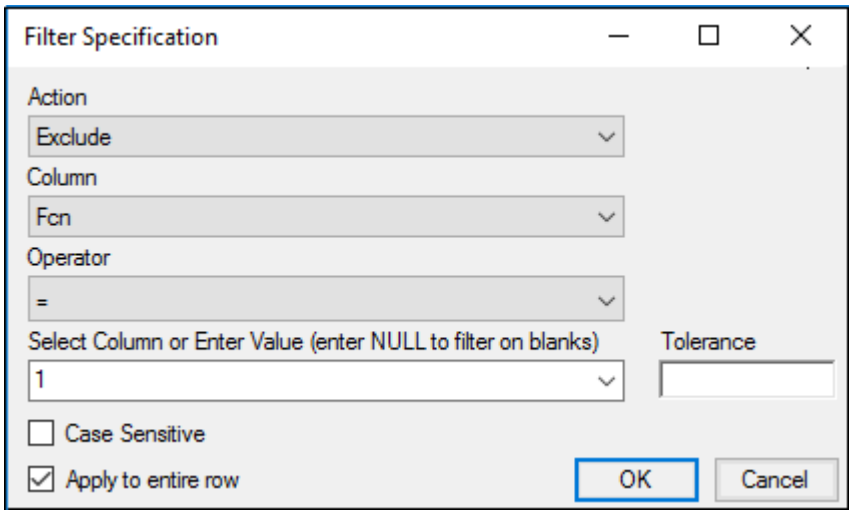
Set up the Maximum Likelihood Models object

1. Create a new project called `PKPD Linked`.
2. Import the dataset …`\Examples\NLME\Supporting files\pkpd.dat`.
3. Click **Finish** in the *File Import Wizard* dialog.

Filter the effect data

The dataset contains drug concentration and effect data. In order to use the dataset with a linked PK/PD model, the concentration and effect data must be filtered into separate columns, and a column for dosing data must be appended to the final worksheet.

1. Right-click **Workflow** in the Object Browser and select **New > Data Management > Data Wizard**.
2. Rename the Data Wizard object `Filter Effect Data`.
3. From the **Action** menu in the Options tab, select **Filter**.
4. Click **Add**.
   The Options tab is populated with tools for defining the filtering criteria to use. The Mappings panel becomes available in the Setup tab and Step 1-No Action is changed to Step 1-Filter in the Setup tab list.
5. Use the mouse pointer to drag the **pkpd** worksheet from the Data folder to the Main Mappings panel.
6. Use the Specify Filter area of the Options tab to specify a filter for the worksheet. The **Built In** filter option button is selected by default.
7. Click **Add** to the left of the **Built In** option.
8. In the *Filter Specification* dialog, leave **Exclude** selected in the **Action** menu.
9. In the **Column** menu select **Fcn**.
10. In the **Select Column or Enter Value** field type `1`.
11. Leave the **Apply to entire row** box checked.



12. Click **OK**.
13. Click (**Execute** icon) to execute the object.

Filter the concentration data

1. Insert a second **Data Wizard** object into the project.
2. Rename the Data Wizard object `Filter Conc Data`.
3. From the **Action** menu in the Options tab, select **Filter**.
4. Click **Add**.
5. Use the mouse pointer to drag the **pkpd** worksheet from the Data folder to the Mappings panel.
6. Use the Options tab to specify a filter for the worksheet. The **Built In** filter option button is selected by default.
7. Click **Add** to the left of the **Built In** option.
8. In the **Action** menu leave **Exclude** selected.
9. In the **Column** menu select **Fcn**.
10. In the **Value** field type 2.
11. Leave the **Apply to entire row** box checked.
12. Click **OK**.
13. Execute the object.

Merge and transform the filtered worksheets

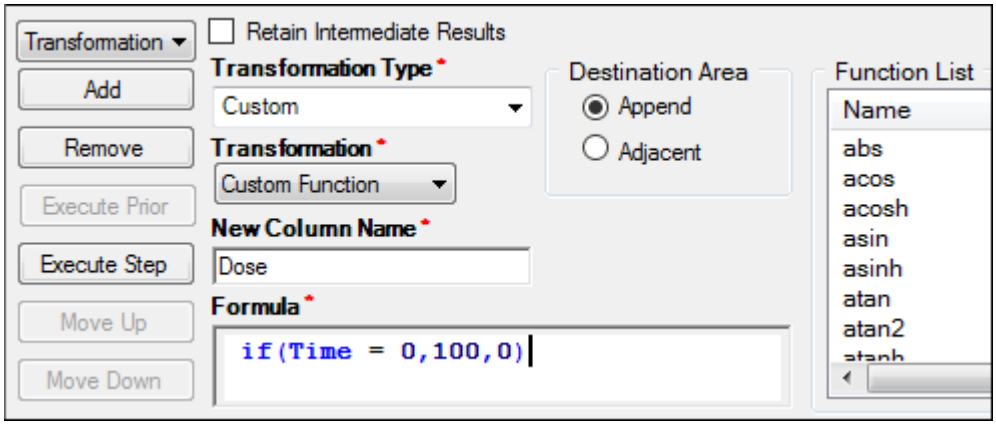
1. Select the workflow in the Object Browser and then select **Insert > Data Management > Merge Worksheets**.
2. In the Worksheet 1 Mappings panel click (**Select Source** icon).
3. In the dialog, select the Filter Effect Data **Result** worksheet and click **OK**.
4. Select the option buttons in the Worksheet 1 Mappings panel to map the data types to the following contexts:

ID to the **Sort** context.
Time to the **Sort** context.
Leave **Fcn** mapped to **None**.
DV to the **Included Column** context.

5. Select **Worksheet 2** in the Setup list.
6. In the Worksheet 2 Mappings panel click the **Select Source** icon.
7. Select the Filter Conc Data **Result** worksheet and click **OK**.
8. Select the option buttons in the Worksheet 2 Mappings panel to map the data types to the following contexts:
   ID to the **Sort** context.
   Time to the **Sort** context.
   Leave **Fcn** mapped to **None**.
   DV to the **Included Column** context.
9. Execute the object.

Add a dosing data column

1. Select the workflow in the Object Browser and then select **Insert > Data Management > Data Wizard**.
2. Rename the Data Wizard object `Column Transform`.
3. From the **Action** menu in the Options tab, select **Transformation**.
4. Click **Add**.
   The Options tab is populated with tools for defining the transformation to take place. The Mappings panel becomes available in the Setup tab and Step 1-No Action is changed to Step 1-Transformation in the Setup tab list.
5. In the Column Transform Main Mappings panel click the **Select Source** icon.
6. Select the Merge Worksheets **Result** worksheet and click **OK**.
7. In the Options tab, select **Custom** from the **Transformation Type** menu.
8. In the **New Column Name** field, type `Dose`.
9. In the **Formula** field, type `if(Time=0,100,0)`.



10. In the Main Mappings panel, make sure that **Time** is mapped to the **Time** context. Leave all other data types mapped to **None**.
11. Execute the object.

12. Right-click the **Result** worksheet in the Results tab list and select **Copy to Data Folder**.

The worksheet is copied to the Data folder and renamed "Result from Column Transform".

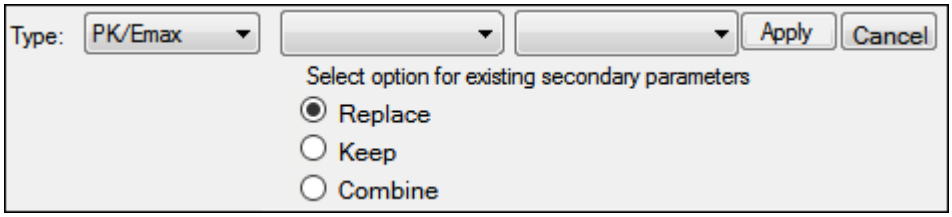
Modify the transformed dataset

1. In the Data folder, rename the "Result from Column Transform" worksheet as `PKPD Data`.
2. In the Columns tab, below the table of data, select the **DV** column header in the Columns list.
3. Click **DV** again to make it editable and type `Effect`.
4. In the same way, rename the **DV_1** column header `Conc`.

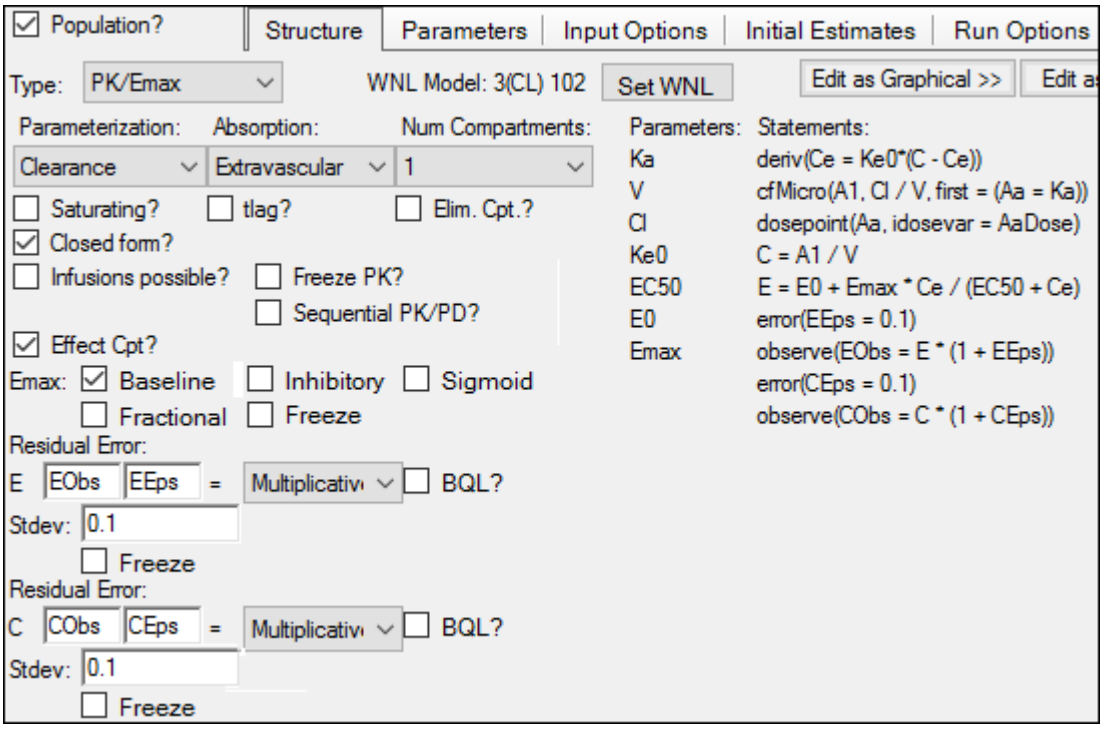
Set up the linked model

The Maximum Likelihood Models interface allows users to select compiled PK, PD, or linked PK/PD library models.

1. Select the **PKPD Data** worksheet in the Data Folder.
2. Right-click the worksheet and select **Send To > Modeling > Maximum Likelihood Models**.
3. In the Structure tab, select **PK/Emax** in the **Type** menu.
4. Click **Set WNL Model**.
   There are two unlabeled menus. The first allows users to select a PK model, and the second allows users to select a PD model.



5. In the PK menu (the first unlabeled menu) select Model 3 (**3 1cp extravascular**).
6. Check the **CL/V** checkbox.
7. In the PD menu (the second unlabeled menu) select Model 102 (**102 emax base**).
8. Click **Apply**. The model is set.
9. Clear the **Freeze PK?** checkbox to add a continuous observation to the model.
10. In the **E** error menu, select **Multiplicative**.
11. In the **C** error menu, select **Multiplicative**.



12. Select the **Parameters > Structural** sub-tab.
13. Clear the **Ran** checkboxes for **Ka** and **E0**.
14. Select the **Fixed Effects** sub-tab.
15. Check the box for **tvKa** to “freeze” the value.
16. Some of the initial estimates must be changed in order for the model to run correctly.
    In the Initial field for **tvKe0**, type `10`.
    In the Initial field for **tvE0**, type `50`.
    In the Initial field for **tvEmax**, type `100`.
17. Select the **Random Effects** sub-tab.
18. Type `0.1` in the **Initial Estimates** fields for nEC50, nEmax, nKe0, nV, and nCl.
19. Select the **Run Options** tab.
20. In the **Algorithm** menu, select **FOCE L-B**.

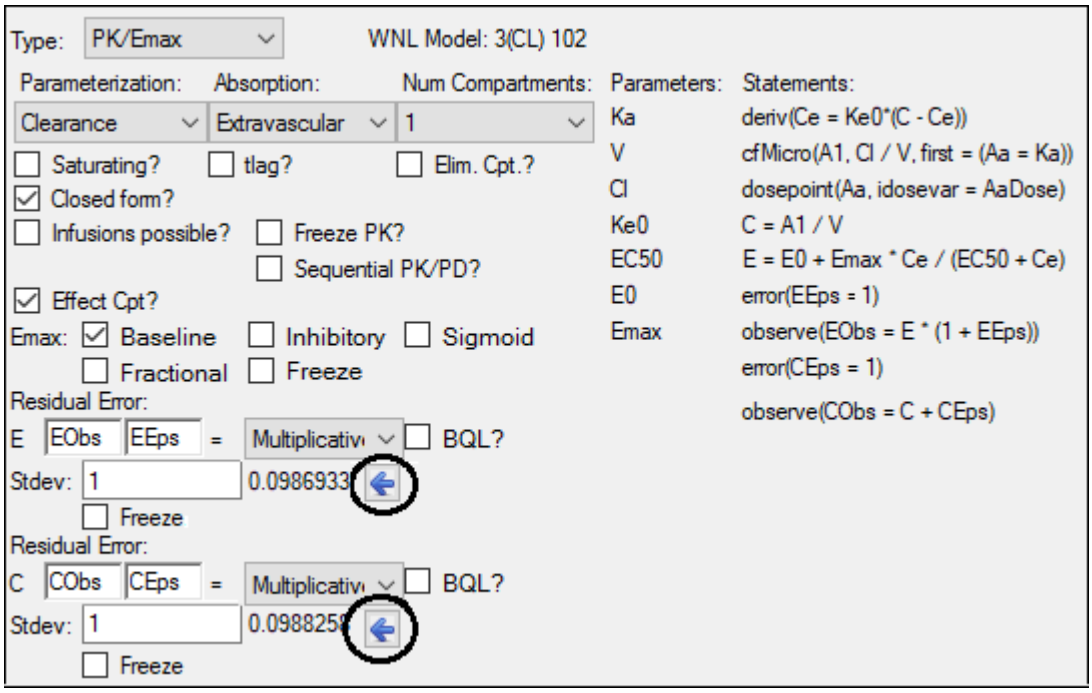
Map the model variables

1. Select the option buttons in the Main Mappings panel to map the data types as follows:
   ID to the **ID** context.
   Time to the **Time** context.
   Effect to the **EObs** context.
   Conc to the **CObs** context.
   Dose to the **Aa** context.
2. Execute the object.

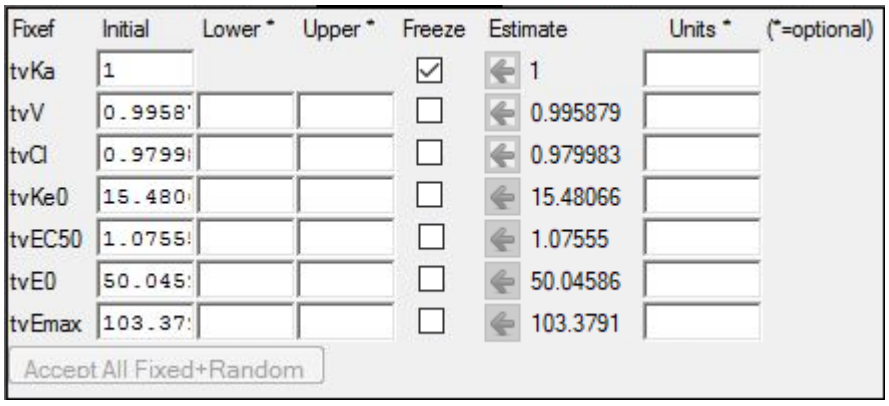
Simulate a population model

This part of the example uses the previously created model to run a simulated model by using the Predictive Check run mode option to perform a Monte Carlo population PK/PD simulation.

1. In the Object Browser, right-click the **ML Model** object and select **Copy**.
2. Right-click the **Workflow** object and select **Paste**.
3. Rename the object as `PKPD Simulation`.
4. In the **Structure** tab, click the left arrow button beside the ***effect*** observation Stdev field to use the new value as the standard deviation.
5. Click the left arrow button beside the concentration ***observation*** Stdev field to use the new value as the standard deviation.



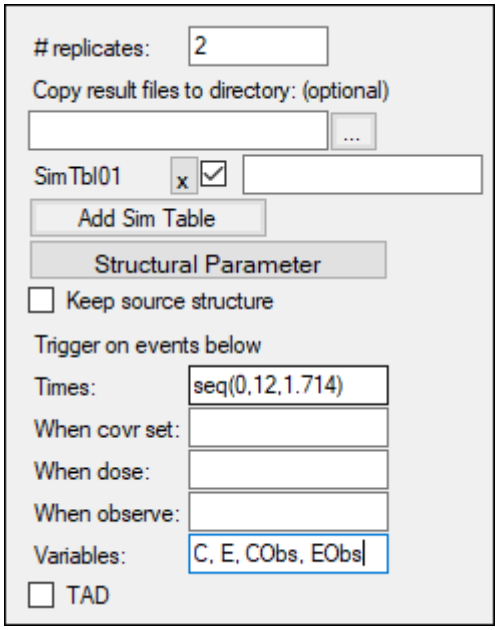
6. Select the **Parameters > Fixed Effects** sub-tab.
7. Click **Accept All Fixed+Random** to copy the model estimates to the **Initial** estimate field for each fixed effect parameter.



The values entered in the Fixed Effects sub-tab are used for all subjects. Because the input dataset contains multiple subjects, the parameter values are needed to create variations in the simulated data.

8. Select the **Run Options** tab.
9. Select the **Simulation** run mode.
10. In the Simulation Options area, click **Add Sim Table** to add a simulation table.
11. In the **Times** field, enter `seq(0,12,1.714)`. This specifies a sequence of times from zero to 12 incremented by 1.714 time units (0, 1.714, 3.428, 5.142, …, 12).
12. In the **Variables** field, enter `C, E, CObs, EObs`.

    C and E are simulated individual predicted values (simulated IPREDs), and CObs and EObs are simulated observations.



13. Execute the object.

Simulation output

A simulation worksheet called PredCheckAll is created. It contains the predictive check simulated data points. All other results listed in the Results tab correspond to the model fit because Phoenix fits the model before performing simulations.

Save and close the project

1. Select **File > Save Project**.
2. Click **Save**.
3. Select **File > Close Project**.

The project is saved and closed and Phoenix can be safely closed.

This concludes the fitting example.

Variations in models using Propofol data

The data in `propofol.dat` follows a three compartment pharmacokinetic model with IV infusion input. This example uses a model similar to the compiled Least-Squares Regression PK model number 19.

Note: The completed project (`Propofol.phxproj`) is available for reference in …`\Examples\NLME`.

Set up the Maximum Likelihood Models object

1. Create a new project called `Propofol`.
2. Import the dataset …`\Examples\NLME\Supporting files\propofol.dat`.
   Click **Finish** in the *File Import Wizard*.
3. Right-click the **propofol** worksheet in the Data folder and select **Send To > Modeling > Maximum Likelihood Models**.
4. In the Structure tab of the ML Model object, select **Macro1** from the **Parameterization** menu.
5. In the **Num Terms** menu, select **3**.
6. Check the **Infusions possible?** checkbox.
7. In the **Residual Error** model menu, select **Multiplicative**.
8. Type `0.3` in the **Stdev** field.
9. Select the **Parameters > Fixed Effects** sub-tab.
10. In the **Initial** column, enter the following initial estimates for each of the study parameters:
    tvV = 11
    tvAlpha = 0.38
    tvB = 0.07
    tvBeta = 0.03
    tvC = 0.004
    tvGamma = 0.002
11. Select the **Random Effects** sub-tab.
12. Type `0.1` in the **Initial Estimates** fields for nV, nAlpha, nB, nBeta, nC, and nGamma.
13. Select the **Run Options** tab.
14. In the **Algorithm** menu, select **Naive pooled**.
15. Select **Forward Diff** in the **Stderr** menu.

Map the model variables

1. Select the option buttons in the Main Mappings panel to map the data types as follows:
   TID to the **ID** context.
   ID to the **None** context. ***ID cannot be mapped to the ID context***.
   TIME to the **Time** context.
   DV to the **C1Obs** context.
   RATE to the **A Rate** context.
   AMT to the **A** context.
   All other data types mapped to **None**.

2. Click  (**Execute** icon) to execute the object.

Accept the new initial estimates

1. Select the **Parameter > Fixed Effects** sub-tab.
2. Click **Accept All Fixed+Random** to copy the new estimates to the Initial estimates field for each parameter.
3. Select the **Run Options** tab.
4. In the **Algorithm** menu, select **FOCE L-B**.
5. Execute the object.

Save and close the project

1. Select **File > Save Project**.
2. Click **Save**.
3. Select **File > Close Project**.

The project is saved and closed and Phoenix can be safely exited.

This concludes the model variations using propofol data example.

Logistic regression modeling in Phoenix

The data in this dataset were simulated to demonstrate logistic modeling. Eight subjects were simulated. Within each subject, the response was randomly generated as either zero or one with probability p, where ln(p/(1 – p))=a+bX+eta as X varies between one and 12.

Note: The completed project (`Logistic_Model.phxproj`) is available for reference in …`\Examples\NLME`.

Set up the Maximum Likelihood Models object

1. Create a new project called `Logistic Model`.
2. Import the dataset …`\Examples\NLME\Supporting files\logistic.dat`.
   Click **Finish** in the *File Import Wizard* dialog.
3. Right-click the worksheet and select **Send To > Modeling > Maximum Likelihood Models**.
4. In the Structure tab of the ML Model object, select **Linear** from the **Type** menu.
5. In the **Linear** menu, select **E = Alpha + Beta*C**.
6. Select the **Parameters > Structural** sub-tab.
7. Clear the **Ran** checkbox beside **Beta** to remove random effects from the Beta parameter.

Map the model variables

1. Select the option buttons in the Main Mappings panel to map the data types as follows:
   ID to the **ID** context.
   rep to the **C** context.
   response to the **EObs** context.

Change the built-in model to a text model

1. Click **Edit as Textual**.
2. In the confirmation dialog, click **Yes**.
3. Select **Model** in the Setup list.

The model text is displayed in the Model text panel as follows:

```
test(){
     covariate(C)
     E = Alpha + Beta*C
     error(EEps = 1)
     observe(EObs(C) = E + EEps)
     stparm(Alpha = tvAlpha * exp(nAlpha))
     stparm(Beta = tvBeta)
     fixef(tvAlpha = c(, 1, ))
     fixef(tvBeta = c(, 1, ))
     ranef(diag(nAlpha) = c(1))
}
```

Edit the text model

The model text needs to be edited to update the error and observation statements.

Note: Users can either type in the Model panel to edit the model text or copy and paste the lines of code from this example.

1. Change the error statement to a probability statement:
   ```
   error(EEps = 1)
   ```
   to
   ```
   prob = exp(E)/(1 + exp(E))
   ```
2. Change the observation statement to a log likelihood statement:
   ```
   observe(EObs(C) = E + EEps)
   ```
   to
   ```
   LL(EObs, EObs == 1 ? log(prob):log(1 - prob))
   ```
3. Click  (**Execute** icon) to execute the object.

Use the multi statement

Edit the model text to use the multi statement, rather than the LL statement, for the multinomial response. The multi statement can be used for multinomial responses from two to ten ordered categories, and allows use of a variety of link functions in addition to the ilogit function (inverse logit).

1. Right-click **ML Model** in the workflow and select **Copy**.
2. Right-click the **Workflow** object and select **Paste**.
3. Rename the model copy as `ML Model_multi`.
4. In the Model text panel, change the two statements
   ```
   prob=exp(E)/(1 + exp(E))
   ```
   and
   ```
   LL(EObs, EObs == 1 ? log(prob):log(1 - prob))
   ```
   to one statement
   ```
   multi(EObs, ilogit, E)
   ```
5. Execute the object.

Save and close the project

1. Select **File > Save Project**.
2. Click **Save**.
3. Select **File > Close Project**.
4. The project is saved and closed and Phoenix can be safely exited.
5. This concludes the logistic regression modeling example.

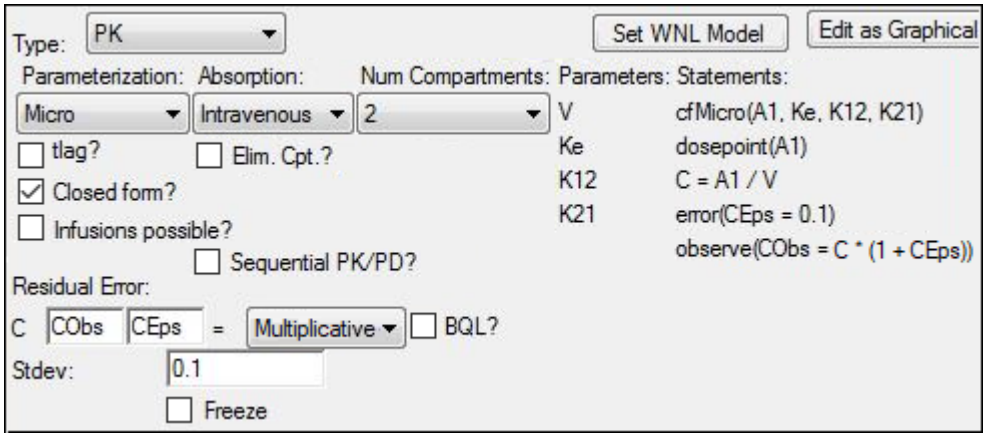
Graphically create a peripheral elimination model

This example shows how to turn a standard two-compartment model into a model with non-standard elimination by using the graphical editor.

Note: The completed project (`Periph Elim.phxproj`) is available for reference in …`\Examples\NLME`.

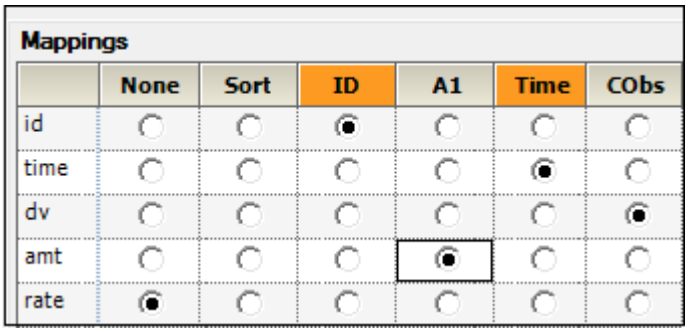
Set up the Maximum Likelihood Models object

1. Create a new project called `Periph Elim`.
2. Import the dataset …`\Examples\NLME\Supporting files\peripheral elim.dat`. Click **Finish** in the *File Import Wizard* dialog.
3. Right-click the worksheet and select **Send To > Modeling > Maximum Likelihood Models**.
4. In the **Structure** tab of the Phoenix model, select **Micro** from the **Parameterization** menu.
5. In the **Num Compartments** menu, select **2**.
6. Change C (continuous observation) to a **Multiplicative** error model.
7. The **Stdev** field should read `0.1`.



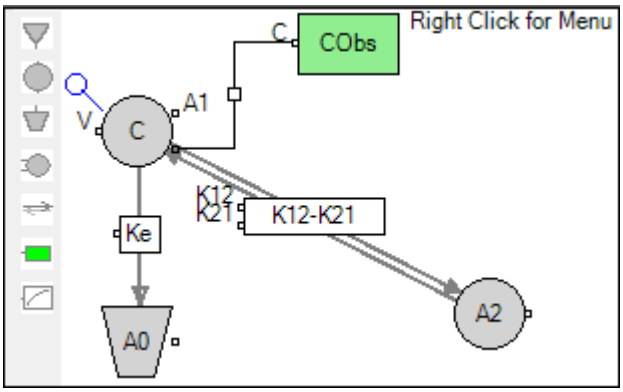
Map the model variables

1. Select the option buttons in the Main Mappings panel to map the data types as follows:
   id to the **ID** context.
   time to the **Time** context.
   dv to the **CObs** context.
   amt to the **A1** context.
   Leave **rate** mapped to **None**.



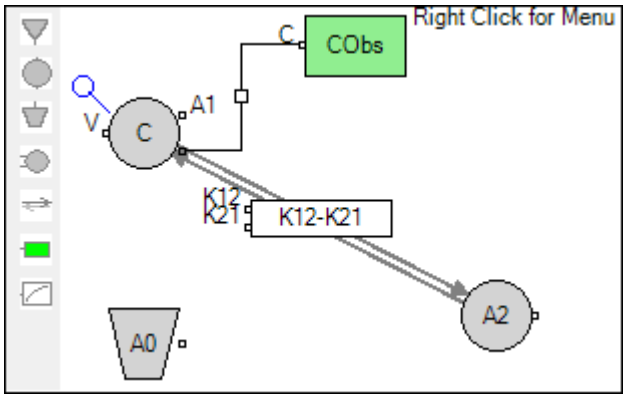
Edit the graphical model

1. Click **Edit as Graphical**.
   In the confirmation dialog, click **Yes**.
   In a second confirmation dialog about not using the closed-form, click **Yes**.
2. If needed, click **Model** in the Setup tab list to display the Model diagram panel.



3. Delete the PK flow between the Central compartment **C** and the Elimination compartment **A0** by selecting the square labeled **Ke**, then right-clicking and selecting **Delete**.
4. Confirm the deletion by clicking **Yes** in the dialog.

   The graphical model now looks like this:



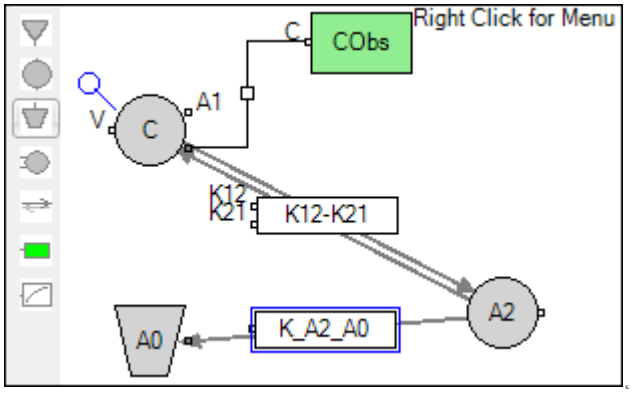
5. Add a PK flow between the Peripheral compartment **A2** and the Elimination compartment **A0**.
   A PK flow can be added in one of two ways:

   Click ⇄ in the Maximum Likelihood Models object toolbox.
   Or

Right-click anywhere in the Model diagram panel and select **Insert > Flow**.

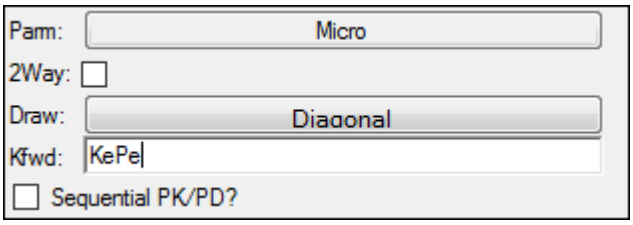
When the PK flow is inserted, the first and second compartments of the flow must be selected.
Left-click the first compartment of the flow, the Peripheral compartment **A2**.
Left-click the second compartment of the flow, the Elimination compartment **A0**.

The PK flow is inserted between the two-compartments.

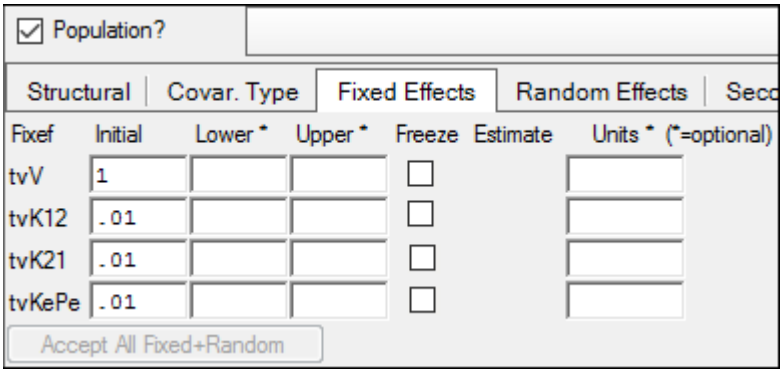


6. Select the PK flow named **K_A2_A0**, if it is not already selected, and type KePe in the Structure tab field **Kfwd**.

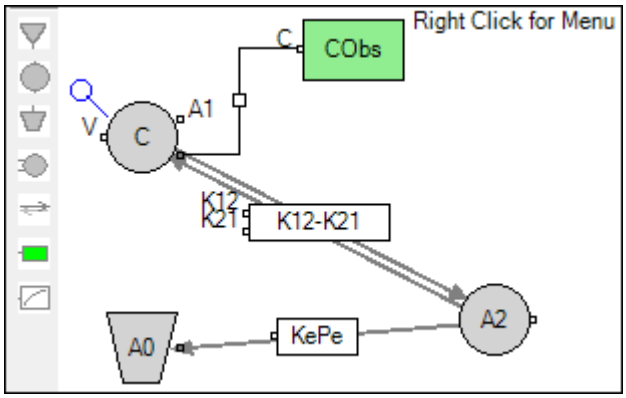
   KePe stands for the rate of elimination between the peripheral compartment and the elimination compartment.



7. Select the **Parameters > Fixed Effects** sub-tab.
8. In the **Initial** column, type the following initial estimates for each of the parameters:
   tvV = 1
   tvK12 = .01
   tvK21 = .01
   tvKePe = .01



9. Select the **Run Options** tab.
10. In the **Algorithm** menu select **Naive pooled**.



11. Click (**Execute** icon) to execute the object.
    The first model execution is used to find better initial estimates for the fixed effects.
12. Select the **Parameters > Fixed Effects** sub-tab.
13. Click **Accept All Fixed+Random** to copy the new estimates to the Initial estimates field for each parameter.
14. Select the **Run Options** tab.
15. In the **Algorithm** menu select **FOCE L-B**.
16. Execute the object.

Save and close the project

1. Select **File > Save Project**.
2. Click **Save**.
3. Select **File > Close Project**.

The project is saved and closed and Phoenix can be safely exited.

This concludes the peripheral elimination model example.

Model using logarithmic concentration values (Lyon dataset)

The data used in the Lyon dataset is from a simulated trial. The concentration data is the logarithm of the concentration data. In this model the logarithm of each prediction value is used to predict the concentration values.

Note: The completed project (`Lyon05.phxproj`) is available for reference in …`\Examples\NLME`.

Set up the Phoenix model object

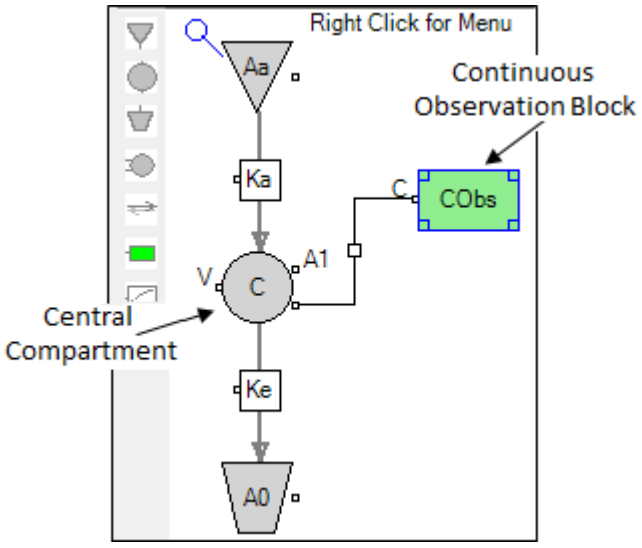
1. Create a new project called `Lyon05`.
2. Import the dataset …`\Examples\NLME\Supporting files\lyon05dose.dat`.
   Click **Finish** in the *File Import Wizard* dialog.
3. Right-click the **lyon05dose** worksheet and select **Send To > Modeling > Maximum Likelihood Models**.
4. In the Structure tab, select **Micro** from the **Parameterization** menu.
5. In the **Absorption** menu, select **Extravascular**.

Map the model variables

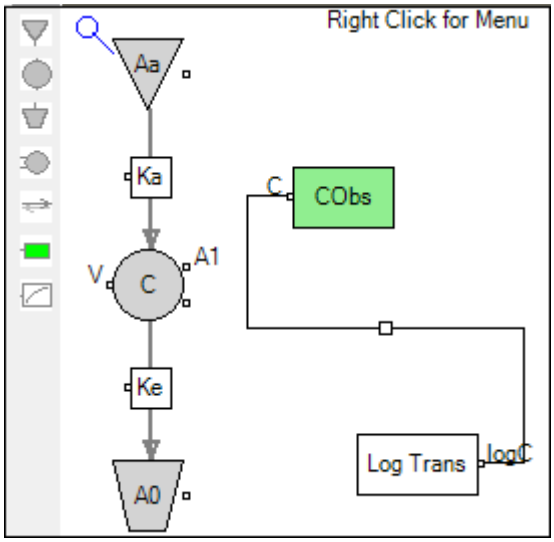
1. Select the option buttons in the Main Mappings panel to map the data types as follows:
   xid to the **ID** context.
   time to the **Time** context.
   dose to the **Aa** context.
   yobs to the **Cobs** context.

Edit the graphical model

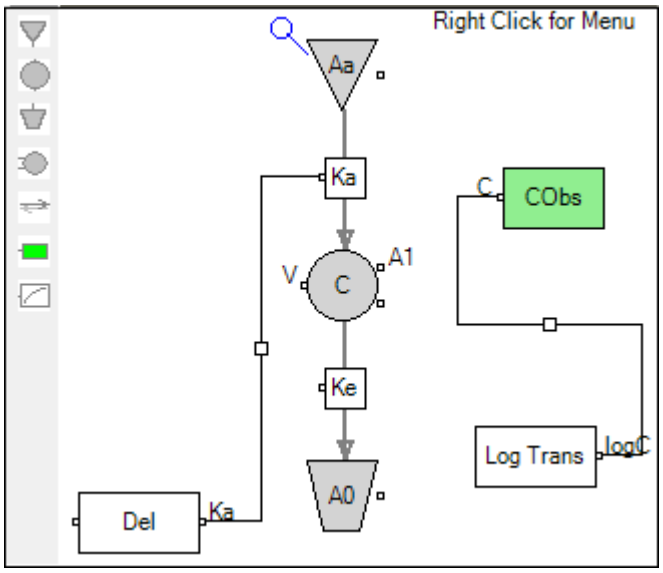
1. Click **Edit as Graphical**.
   In the confirmation dialog, click **Yes**.
   In a second confirmation dialog about not using the closed-form, click **Yes**.
2. If needed, click **Model** in the Setup tab list to display the Model diagram panel.



3. Delete the connector between the Continuous Observation block **CObs** and the Central compartment **C**.
   Left-click the square in the middle of the connector to select it.
   Right-click the square and select **Delete**.
   Confirm the deletion by clicking **Yes** in the dialog.
4. Add a **Procedure** block by right-clicking anywhere in the Model diagram panel and select **Insert > Procedure**.
   A Procedure block named Proc1 is added to the Model diagram.
5. In the **Name** field in the Structure tab, type `Log Trans`.
6. In the **Code** sub-tab of the Structure tab, type `logC=log(C)` in the **Code** field.
7. Left-click the Log Trans output square, labeled **logC**, and drag it to the input square of the Continuous Observation block, labeled **C**.

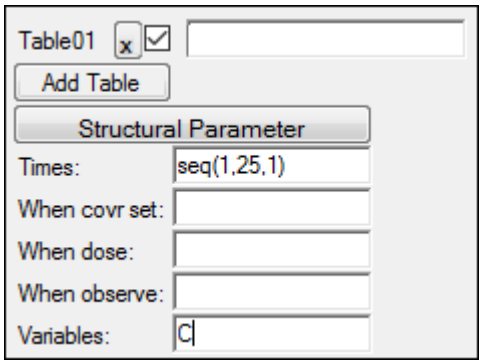


8. Right-click anywhere in the Model diagram panel and select **Insert > Procedure** to add a second Procedure block.
9. In the **Name** field, type `Del`.
10. In the **Code** sub-tab, type `Ka=Ke+Del` in the **Code** field.
11. Select the **Str. Parameters** sub-tab and click **Add**.
12. In the field, type `Del`.
13. In the Model diagram panel, left-click the Del output square, labeled **Ka**, and drag it to the input square of Ka PK flow, labeled **Ka**.



Note: Connecting Del's Ka output to PK flow Ka's input changes Ka from a structural parameter to an output parameter.

14. Select the **Parameters > Secondary** sub-tab.
15. Click **Add**.
16. Type `tvKa` in the **Parameter** field.
17. Type `tvKe+tvDel` in the **Definition** field.
18. Select the **Random Effects** sub-tab.
19. Clear the **Diag** checkbox to use a full block structure.
20. Select the **Fixed Effects** sub-tab.
21. Enter the following initial estimates:
    tvV = 25
    tvKe = 0.3
    tvDel = 0.3
22. Select the **Run Options** tab.
23. Select **FOCE L-B** as the algorithm.
24. Select **Add Table**. The table definition fields are displayed.
25. In the **Times** field, type `seq(1,25,1)`.
26. In the **Variables** field, type `C`.



27. Click (**Execute** icon) to execute the object.

Run the model using a different engine

Although the FOCE L-B model engine was used, the Lyon data is also valid for use with the QRPEM model engine. As a continuation of this example, create a copy of the Maximum Likelihood Models object and use the QRPEM method to generate results.

1. Right-click the **ML Model** object in the Object Browser and select **Copy**.
2. Right-click on the **Workflow** object in the Object Browser and select **Paste**.
3. Rename the **Copy of ML Model** object to `QRPEM Model`.

   All of the mappings and other settings are retained in the QRPEM Model object, so the only change needed is to select QRPEM as the engine.
4. Click the **Run Options** tab.
5. Select **QRPEM** from the **Algorithm** menu.
6. Execute the object.

Save and close the project

1. Select **File > Save Project**.
2. Click **Save**.
3. Select **File > Close Project**.

The project is saved and closed and Phoenix can be safely exited.

This concludes the modeling with logarithmic concentration values example.

Creating a two dose point model for intravenous and oral data

This example shows how to turn a single dose point model into a two dose point model using the graphical editor.

The data used in the cmt1ivpo dataset is from a simulated trial. The purpose of this example is to show how to use a dataset that contains multiple dosing routes with the Phoenix NLME GUI.

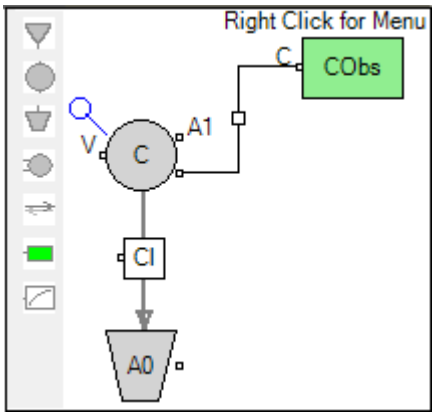
Note: The completed project (`IVPO.phxproj`) is available for reference in …`\Examples\NLME`.

Set up the Maximum Likelihood Models object

1. Create a new project called `IVPO`.
2. Import the dataset …`\Examples\NLME\Supporting files\cmt1ivpo.dat`.
   Click **Finish** in the *File Import Wizard* dialog.
3. Right-click the **cmt1ivpo** worksheet and select **Send To > Modeling > Maximum Likelihood Models**.
4. In the **Structure** tab of the Phoenix model, select **Clearance** from the **Parameterization** menu, if needed.
5. In the **C** (Continuous Observation) error model menu, select **Multiplicative**.
6. Keep the default value of `0.1` in the **Stdev** field.

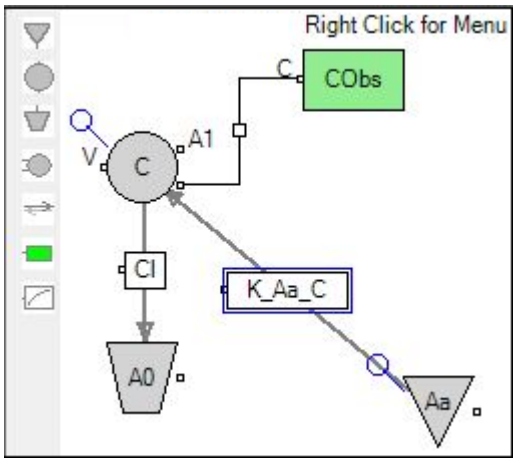
Graphically edit the model

1. Click **Edit as Graphical**.
   In the confirmation dialog, click **Yes**.
   In the second confirmation dialog about not using the closed-form, click **Yes**.
2. Select **Model** in the Setup tab list, if needed.



3. Add an **Absorption** compartment by clicking  in the Maximum Likelihood Models object toolbox.
4. Add a PK flow between the Absorption compartment **Aa** and the Central compartment **C** by clicking  in the toolbox.
   Click the first compartment of the flow, the Absorption compartment **Aa**.
   Click the second compartment of the flow, the Central compartment **C**.

The PK flow is inserted between the two-compartments and is labeled **K_Aa_C**.



5. Select the **Parameters > Fixed Effects** sub-tab.
6. In the **Initial** column, type `0.2` as the initial estimate for **tvCl**.
7. Select the **Random Effects** sub-tab.
8. Type `0.1` in the fields for nV, nCl, and nK_Aa_C.

Map the model variables

1. Select **Main** in the Setup list.
2. Use the option buttons in the Mappings panel to map the data types as follows:
   id to the **ID** context.
   time to the **Time** context.
   conc to the **CObs** context.
   doseiv to the **A1** context.
   dosepo to the **Aa** context.
3. Click (**Execute** icon) to execute the object.

Save and close the project

1. Select **File > Save Project**.
2. Click **Save**.
3. Select **File > Close Project**.

The project is saved and closed and Phoenix can be safely exited.

This concludes the two dose point model example.

Covariate modeling with the PK01 dataset

The data used in the PK01 dataset is from a simulated trial. The purpose of this example is to show how to add covariate effects to a model to determine which covariates have an effect on the model's results.

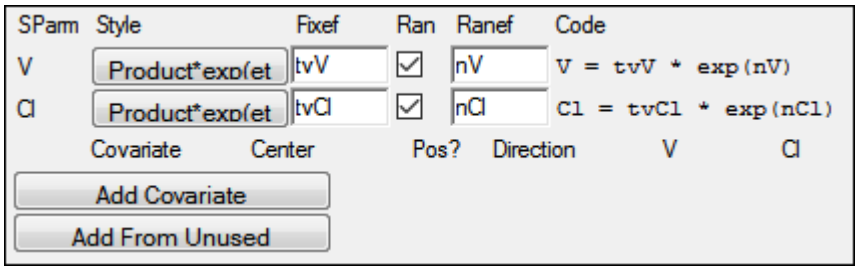
Note: The completed project (`PK01Co3.phxproj`) is available for reference in …`\Examples\NLME`.

Set up the Maximum Likelihood Models object

1. Create a new project called `PK01Co3`.
2. Import the dataset …`\Examples\NLME\Supporting files\pk01cov3.dat`.
   Click **Finish** in the *File Import Wizard* dialog.
3. Right-click the worksheet and select **Send To > Modeling > Maximum Likelihood Models**.
4. Rename the object `Base Model`.
5. Select the option buttons in the Main Mappings panel to map the data types to the following contexts:
   id to the **ID** context.
   time to the **Time** context.
   conc to the **CObs** context.
   dose to the **A1** context.
   Leave all other data types mapped to **None**.
6. In the **Structure** tab, make sure **Clearance** is selected in the **Parameterization** menu.
7. In the **C** (Continuous Observation) error model menu, select **Multiplicative**.
8. Select the **Parameters > Fixed Effects** sub-tab.
9. In the **Initial** column, type the following initial estimates for each of the study parameters:
   tvV 10
   tvCl 0.1

Add covariates

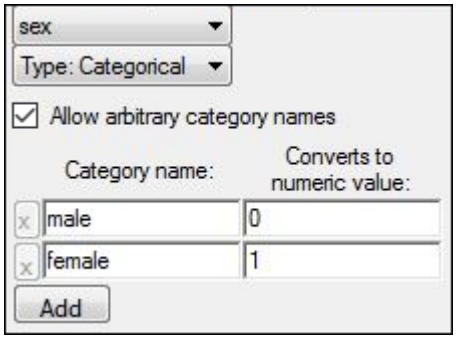
1. Select the **Structural** sub-tab.



2. Click **Add From Unused**.
3. Select the checkboxes beside **sex**, **wt**, and **age**.
4. Click **Add**.
   The covariate data types are automatically mapped to the correct contexts.

	None	Sort	ID	Aa	Time	CObs	sex	wt	age
id	○	○	●	○	○	○	○	○	○
time	○	○	○	○	●	○	○	○	○
conc	○	○	○	●	○	○	○	○	○
dose	○	○	○	○	○	●	○	○	○
sex	○	○	○	○	○	○	●	○	○
wt	○	○	○	○	○	○	○	●	○
age	○	○	○	○	○	○	○	○	●

5. Select the **Covar. Type** sub-tab.
   Sex is automatically selected in the **Select covariate** menu.
6. In the **Type** menu, select **Type: Categorical**.
7. Check the **Allow arbitrary category names** checkbox.
8. In the first field underneath **Category Name**, type `male`.
9. Leave `0` as the value for male.
10. In the second field underneath **Category Name**, type `female`.
11. Type `1` in the value field for female.



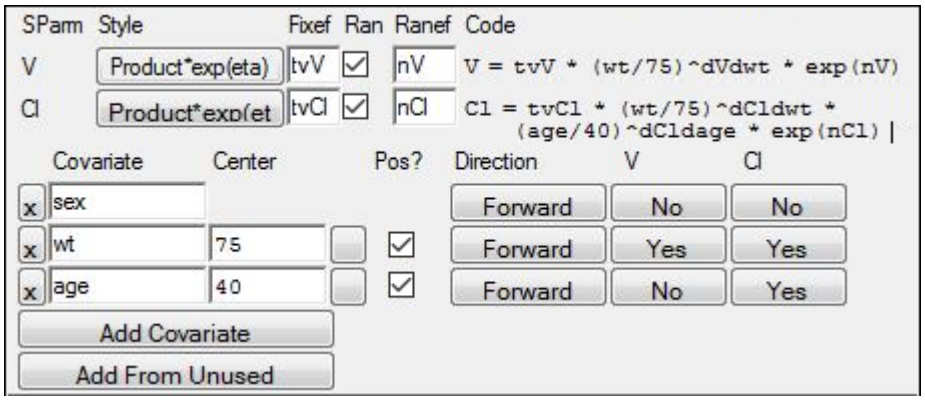
12. Click (**Execute** icon) to execute the object.

Add covariate effects for weight and age

In this model the covariate effects for weight and age are added to the study parameters.

1. Right-click **Base Model** and select **Copy**.
2. Right-click the **Workflow** object and select **Paste**.
   The model object and its settings are pasted into the PK01Co3 model project and named Copy of Base Model.
3. Rename **Copy of Base Model** to `Covariate Model`.
4. In the Main Mappings panel, check that the data types are mapped as follows:
   id to the **ID** context.
   time to the **Time** context.
   conc to the **CObs** context.
   dose to the **A1** context.
   sex, **wt**, and **age** to **sex**, **wt**, and **age**, respectively.

5. Select the **Parameters > Structural** sub-tab.
6. In the **V** column, click **No** beside **wt** to select a covariate effect.
7. In the **Cl** column, click **No** beside **wt** to select a covariate effect.
   The study variable weight is added as a covariate effect to both **V** and **Cl**.
8. In the **Center** field for **wt** type `75`.
9. In the **Cl** column, click **No** beside **age** to select a covariate effect.
   The study variable age is added as a covariate effect to **Cl**.
10. In the **Center** field for **age** type `40`.



11. Select the **Fixed Effects** sub-tab.
12. Click **Accept All Fixed+Random** to copy the new estimates to the Initial estimates field for each parameter.
13. In the **Initial** column, leave the initial values for dVdwt, dCldwt, and dCldage set to `0` (zero).
14. Execute the object.

Add sex as a covariate effect

In this model the covariate effect for sex is added to the study parameter volume.

1. Right-click **Covariate Model** and select **Copy**.
2. Right-click the **Workflow** object and select **Paste**.
3. Rename the copied object as `Covariate Model_sexcov`.
4. In the Main Mappings panel, check that the data types are mapped as follows:
   id to the **ID** context.
   time to the **Time** context.
   conc to the **CObs** context.
   dose to the **A1** context.
   sex, **wt**, and **age** to **sex**, **wt**, and **age**, respectively.
5. Select the **Parameters > Structural** sub-tab.
6. In the **V** column, click **No** beside the **sex** parameter.
   The study variable sex is added as a covariate effect to **V**.
7. Select the **Fixed Effects** sub-tab.
8. Click **Accept All Fixed+Random** to copy the new estimates to the Initial estimates field for each parameter.

9. In the **Initial** column, type 1 for the initial estimate for the parameter **dVdsex1**.
10. Execute the object.
11. In the Results tab, select the **Core Output** text file.

Note that adding sex as a covariate has no effect on the fixed effects values, so the covariate can be removed from the model.

The concludes the covariate modeling example using the PK01 dataset.

Using a linked indirect response model

The data in `link-IPR.dat` consists of measured effect values over time in 10 subjects, following a single dose at time zero. Effect is measured at the same 14 time points in all subjects. A reasonable fit is obtained utilizing a two-compartment model with first-order absorption linked to an indirect response model with inhibition of input, both of which are available as Least-Squares Regression PK model 11 and Indirect Response model 51.

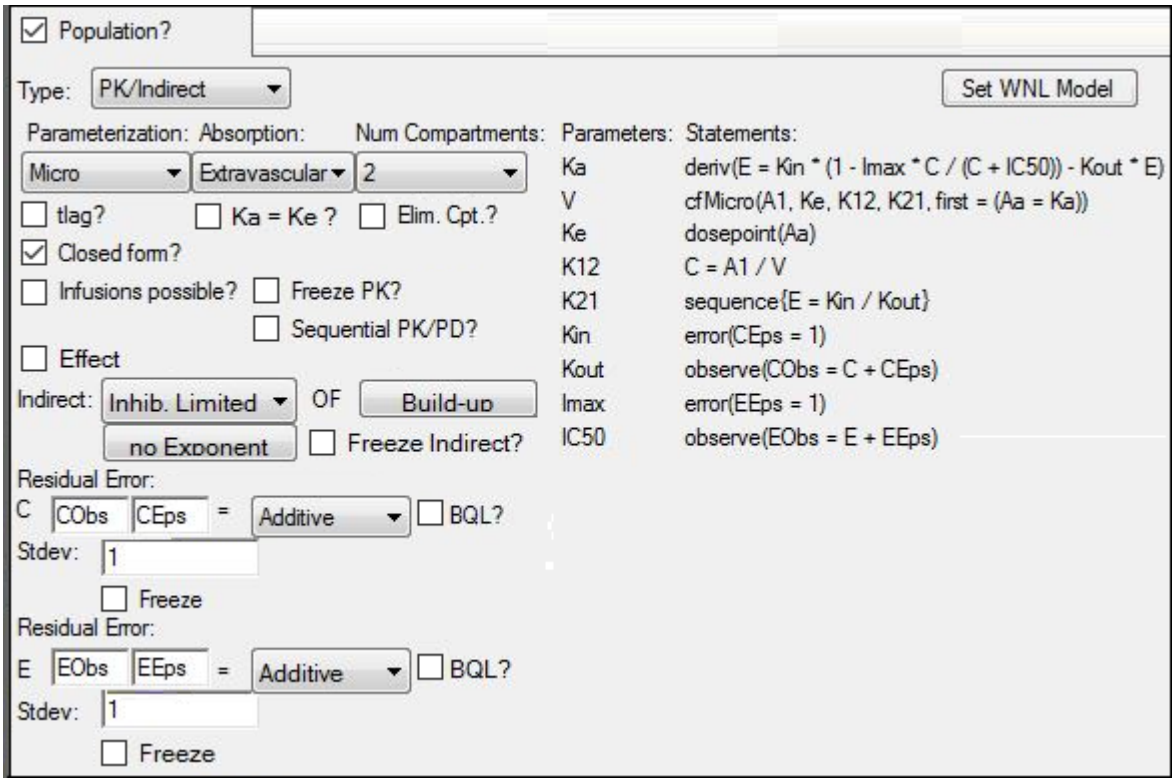
The effect is modeled as a function of the drug concentration in the central compartment, Cp, and parameters representing the measured response to the drug, R, a zero-order constant for production of response, Kin, a first-order rate constant for loss of response, Kout, and the drug concentration at half-maximum inhibition, IC50.

Within-subject error is assumed independently distributed with constant variance.

Note: The completed project (`Linked_IPR.phxproj`) is available for reference in …`\Examples\NLME`.

Set up the Maximum Likelihood Models object

1. Name the new project `Linked IPR`.
2. Import the dataset …`\Examples\NLME\Supporting files\link-IPR.dat`. Click **Finish** in the *File Import Wizard* dialog.
3. Right-click the worksheet and select **Send To > Modeling > Maximum Likelihood Models**.
4. In the Structure tab of the Maximum Likelihood Models object, select **PK/Indirect** from the **Type** menu.
5. Select **Micro** from the **Parameterization** menu.
6. In the **Absorption** menu, select **Extravascular**.
7. In the **Num Compartments** menu, select **2**.
8. In the **Indirect** menu, select **Inhib. Limited**.



9. Select the **Parameters > Structural** sub-tab.
10. Clear the **Ran** checkbox beside **Imax** to remove any random effects from the parameter.
11. Select the **Fixed Effects** sub-tab.
12. Enter the following initial estimates:
    tvKa = 0.92
    tvV = 2.44
    tvKe = 0.44
    tvK12 = 0.36
    tvK21 = 0.24
    tvKin = 400
    tvKout = 2
    tvImax = 1
    tvIC50 = 15
13. Check the **Freeze** checkbox beside **tvImax** to freeze the parameter.
14. Select the **Run Options** tab.
15. In the **Algorithm** menu, select **Naive pooled**.

Map the model variables and dosing

1. Use the option buttons in the Main Mappings panel to map the data types to the following contexts:
   time to the **Time** context.
   subject to the **ID** context.
   effect to the **EObs** context.
2. Click **Dosing** in the Setup tab.
3. In the Dosing Mappings panel, turn on the **Use Internal Worksheet** checkbox.

4. In the table, enter `100` in the first cell of the **Aa** column.
5. Enter `0` in the first cell of the **Time** column.
6. Select the two cells containing the values just entered and drag the selection to the include all rows.
   When the mouse button is released, all rows in the table will have “100” and “0” values for the Aa and Time columns, respectively.
7. Click  (**Execute** icon) to execute the object.
8. In the Results tab, check the **Core Output** text file. The fixed effect value for epsilon (EEps) is used as the standard deviation for the epsilon error model in the next model execution.

Change the model to full block

1. Select the **Structure** tab.
2. Click  beside the **E Stdev** field to use the new value for residual error.
3. Select the **Parameters > Random Effects** sub-tab.
4. Clear the **Diag** checkbox to create a full block model.
5. Select the **Run Options** tab.
6. In the **Algorithm** menu, select **FOCE L-B**.
7. Execute the object.

This concludes the linked indirect response modeling example.

Simple PK population analysis using QRPEM

This example shows how to perform a simple PK population analysis with the QRPEM engine, including dataset importation, model set up, and execution for a simple one-compartment model with parameter Cl (Clearance) and V (Volume).

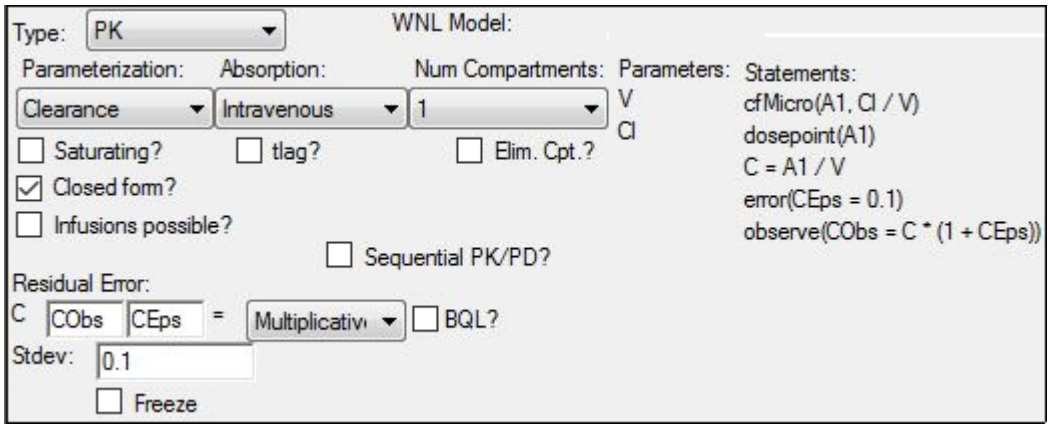
Data and dosing information are assumed already collected for each patient and entered into a data file (`model1pkcl.csv`).

Also assumed available, using either prior information or tools like non-compartmental analysis or graphical tools in Phoenix, are initial values for the population mean of Cl and V. These will be needed by the model.

Note: The completed project (`1c_iv_qrpem.phxproj`) is available for reference in …`\Examples\NLME`.

Set up the Maximum Likelihood Models object

1. Create a new project called `1c_iv_qrpem`.
2. Import the dataset …`\Examples\NLME\Supporting files\model1pkcl.csv`. Click **Finish** in the *File Import Wizard* dialog.
3. Right-click the worksheet and select **Send To > Modeling > Maximum Likelihood Models**.
4. Use the option buttons in the Main Mappings panel to map the data types as follows:
   AMT to the **A1** context
   DV to the **CObs** context
   ID and **Time** are automatically mapped
5. In the **Structure** tab, make sure the **Parameterization** menu is set to **Clearance**, the **Absorption** menu to **Intravenous**, and the **Num** menu to **1**.
6. Set the **Residual Error** menu to **Multiplicative**.
7. Keep the **Stdev** value of **0.1**.



Note: For PK data with an assumed proportional error model, the use of 0.1 (10% error) as initial input for the standard deviation is often reasonable. Since sigma is also optimized, any prior knowledge is beneficial and can prevent potential confounding between the within (sigma) and between (Omega) subject variability model-based parameters.

Enter initial estimates for the population means

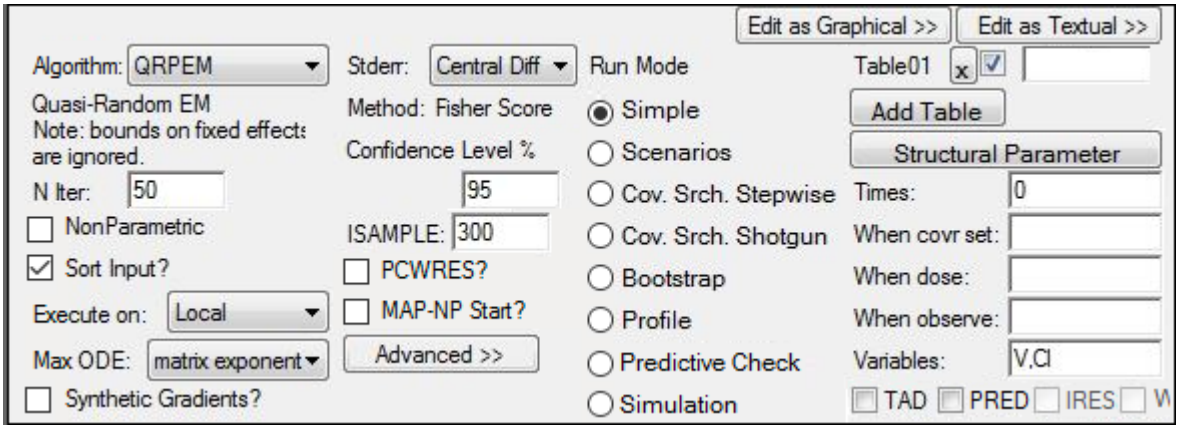
The MCPEM/QRPEM algorithm's first step requires an initial estimate for the population mean (Fixed Effect) and variance (Random Effect). For each patient, given its data, Phoenix NLME will take these population means and variance (and Sigma) as input and will estimate the most likely set of parameters that explain the patient's response data. The algorithm finds the sets of values that could explain the same data and, since each set has a different probability in explaining the patient's data, it defines a distribution (referred to as a posterior distribution).

1. Select the **Parameters > Fixed Effects** sub-tab.
2. Enter the following initial estimates:
   tvV = 50
   tvCl = 5
3. Select the **Random Effects** sub-tab.
4. Clear the **Diag** checkbox to specify a full variance covariance matrix.
5. Enter `1` for all diagonal entries and 0 for off-diagonal.

   Starting with initial estimates for the mean and variance and the number of samples to estimate for each patient, the algorithm estimates the individual posterior distributions for each patient. This is used as the sample at the first iteration.

   The algorithm then updates the mean and population variances and generates new posterior distributions which are used as the new sample for each patient. The iterations continue, using the same algorithm but with new mean, variances, and sigma.

   Usually 50 iterations are enough to reach convergence (i.e., no noticeable change in the means, variances, sigma and log-likelihood). Since the algorithm uses samples, there may be small oscillations in the log-likelihood, but no increasing or decreasing trend once convergence is reached.
6. Select the **Run Options** tab.
7. In the **Algorithm** menu, select **QRPEM**.
8. Enter **50** iterations for **N iter**.
   To obtain the best model parameters for each patient, add a table.
9. Click **Add Table**.
10. Enter `0` for **Times**.
11. Click the **Structural Parameter** button to list the structural parameters in the **Variables** field.



12. Click  (**Execute** icon).
13. In the Results tab, click **Table01** under Output Data.
    Table01 reports the individual Cl and V values for each patient (the mode of each individual posterior distribution).
14. Click **Theta** under Output Data.

    The Theta table shows the final means, variances, and sigma.

Use the MAP

So far in this example, the default method has been used. This method works well when the data are not too rich. The main settings used for the default algorithm include 300 samples for each patient, perform a MAP estimation only at the first iteration, and do a maximum of 50 iterations.

When data are very rich, each patient will be defined most likely by very few Cl and V value sets, with most of the weight on only one set of model parameters. With very rich data, the program can have problems with the default algorithm as the individual posterior variances collapse easily to zero because of there are not enough random samples and because the true posterior variances are very small. When this happens, the estimated posterior variances from iteration two can collapse and it is better to ask the program to do a MAP estimation at each iteration. Increasing the number of samples should also be done.

1. In the Object Browser, right-click the **ML Model** object and select **Copy**.
2. Right-click the **Workflow** item and select **Paste**.
3. Rename the copied object as `MAP`.
4. Select the **Run Options** tab.
5. Enter `1000` in the **ISAMPLE** field.
6. Click **Advanced >>**.
7. Turn on the **MAP Assist** checkbox.
8. Execute the object.

For comparison, run the model using the FOCE ELS and FOCE L-B engines instead of QRPEM. QRPEM has a more efficient sampling algorithm, so the number of samples will need to be increased.

This concludes the simple PK population analysis using QRPEM example.

Three category multinomial model with a covariate

For this example, a completed project will be used as an illustration.

This example shows a three category multinomial model with a covariate, run with three different engines: QRPEM, Laplace, and AdaptiveGaussianQuadrature. This project illustrates how to construct such a model with the simple LL statement. It also demonstrates that the accurate methods (QRPEM and AGQ) get virtually identical and "correct" results, in good agreement with the simulation that created the data set, while the approximate method (Laplacian) gets relatively poor results that are quite different.

Information about the example data

1. Select **File > Load Project** to open an existing Phoenix project.
2. In the *Load Project* dialog, navigate to …`\Examples\NLME`.
3. Select **Multinomial_3cat_with_covariates.phxproj** and click **Open**.

The data set consists of simulated data from a multinomial categorical model, where the observations are either y=0,1, or 2 and represent a category. The probability of an observation being a particular category is a function of four structural parameters th1, th2, th4, and th5 as follows:

Prob(y=0)=ilogit(th1+th4*PER+th5*DOSE)
Prob(y=1)=ilogit(th1+th4*PER+th5*DOSE+th2) – Prob(category=0)
Prob(y= 2)=1 – (Prob(y=1)+Prob(y=0))

Here ilogit(x)=exp(x)/(1+exp(x)) is an increasing function of its argument.

The covariate PER (period) can take the value zero or one and can vary within an individual on different occasions. Similarly, the covariate DOSE ranges from zero to 30 and can vary within an individual.

The four structural parameters have the form th1=tvth1+nh1, th2=tvth2, th4=tvth4, and th5=tvth5, where nh1 is a N(0,20) random effect associated with th1 and the nominal values of the fixed effects are: tvth1=10, tvth2=1.5, tvth4= –5, tvth5= –0.01. (These nominal values, along with the N(0,20) distribution for nh1, were used to simulate the data).

Project workflows and objects

The techniques for constructing such a multinomial model are discussed previously in the example "Logistic regression modeling in Phoenix".

Here the focus is on the very different results from three different types of methods depending on whether the method uses an low or high accuracy likelihood approximation.

There are three workflows in the model, as well as a Model Comparison object. The workflows correspond to running exactly the same model with the QRPEM engine, a Laplacian engine, and an adaptive Gaussian quadrature engine with 21 integration points along the one-dimension random effect nh1 axis.

- The QRPEM method uses 300 quasi-random sample points for its evaluation of the Bayesian posterior integrals, and thus can be considered a high accuracy method.
- The Laplacian engine uses only one integration point and is often a relatively low accuracy method, particularly in cases where the observations are categorical or counts.
- The Adaptive Gaussian Quadrature engine with 21 points placed at particularly informative points is also a high accuracy method.

Note that Laplacian can be considered a special case of AGQ where only one integration point is used. In fact, the only difference between the Laplacian and AGQ21 workflows is that in the **N AGQ** box in the Run Options tab, the value is set to **21** for AGQ21 and to **1** for Laplacian.

Results

The results are summarized in the table below (most values are shown rounded to the nearest hundredth):

___	Nominal	QRPEM	Laplacian	ACQ21
-2LL		2541.93	2299.57	2542.56
tvth1	10.00	9.93	14.82	10.13
tvth2	1.50	1.54	1.88	1.56
tvth4	-5.00	-4.94	-6.71	-4.93
tvth5	-0.01	0.00	-0.01	-0.01
Omega (nth 1)	20.00	20.37	84.95	23.06

These three engines use different methods to approximate -2LL. So -2LL values cannot be used to determine which engine is better than the others. From the table above, we can see that QRPEM and AGQ21 get very good (close to nominal) estimates for all fixed effects, while the Laplacian is significantly poorer on all but Tvth5. Finally, the single random effect parameter is Omega and is estimated very well (close to the nominal value of 20) by QRPEM and AGQ21, whereas the Laplacian estimate is very poor.

Exploring Event, Count, and LL Statements

A series of 4 models (Poisson, zero inflated Poisson, negative binomial, zero inflated negative binomial) have been implemented using the `count` statement and in equivalent form using the `LL` statement (8 models in all). The project containing these models is called `CountLL_ModelExamples.phxproj` and can be found in …`\Examples\NLME`. The PML model code provided in these examples can be modified for application in your projects.

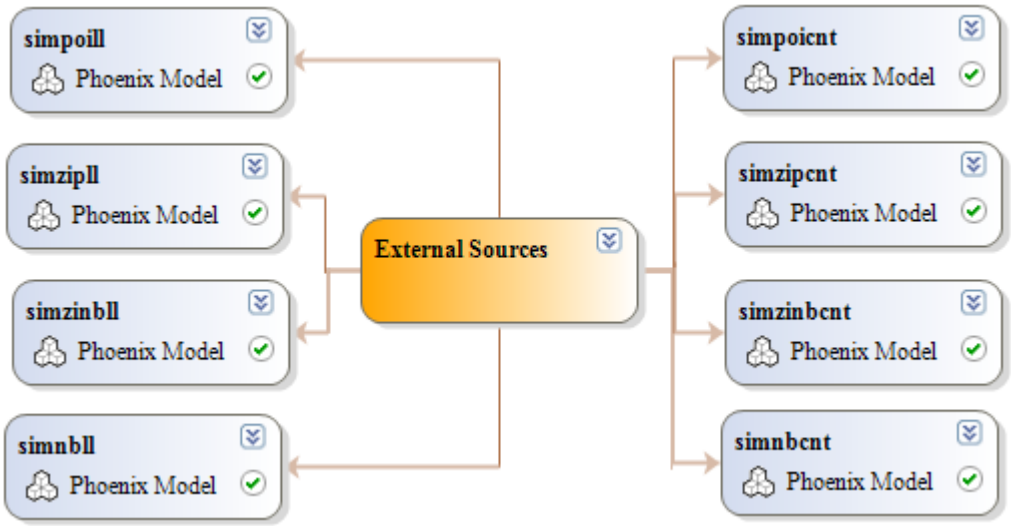
The four datasets used in this example project include:

`simnbnp`: data for the negative binomial model
`simpoinp`: data for the Poisson model
`simzinbnp`: data for the zero inflated negative binomial model
`simzipnp`: data for the zero inflated Poisson model

The eight Maximum Likelihood Models objects include:

simpoicnt: A simple count statement version of a Poisson count model.
simpoill: A simple LL version of Poisson count model.
simzipcnt: A count statement version of zero inflated Poisson count model.
simzipll: An LL statement version of zero inflated Poisson count model.
simzinbll: An LL statement version of zero inflated negative binomial count model.
simzinbcnt: A count statement version of zero inflated negative binomial count model.
simnbcnt: A count statement version of negative binomial count model.
simnbll: An LL statement version of negative binomial count model.

All of these models involve a linear dependence of log λ on a covariate “dose”.



There is some additional information at the top of the PML code for each model. In the Settings tab, click **Model**. The PML code is displayed for viewing/editing.

The example project called `TTEHazard_ModelsExamples.phxproj`, found in …`\Examples\NLME` directory, illustrates the use of the `event` statement in a Time To Event (TTE) hazard context.

There are really two observations

- whether an event happened
- the time stamp of the event (or end of the observation period, if no event occurred), which is taken to be the time of the `cens` observation.

In the example project, the hazard function `haz` (which is a constant in the example) is integrated over the current period by a hidden differential equation `dcumhazard/dt=haz`, which yields a survival probability of `S=exp(-cumhazard)`.

The likelihood (if an event is observed) is `S*haz` and S (if an event is not observed). This likelihood computation and the differential equation computation are automatically invoked by `event(cens,haz)`.

An example of individual modeling with Maximum Likelihood Model object

This example illustrates how a dataset can be fitted to a two-compartment model with first-order absorption in the pharmacokinetic model library using either the Least-Squares Regression PK Model or a Maximum Likelihood Models object GUI.

Exp1 data is used, which is a dataset fit to PK model 13 in the pharmacokinetic model library. Four constants are required for model 13: the stripping dose associated with the parameter estimates, the number of doses, the dose, and the time of dosing.

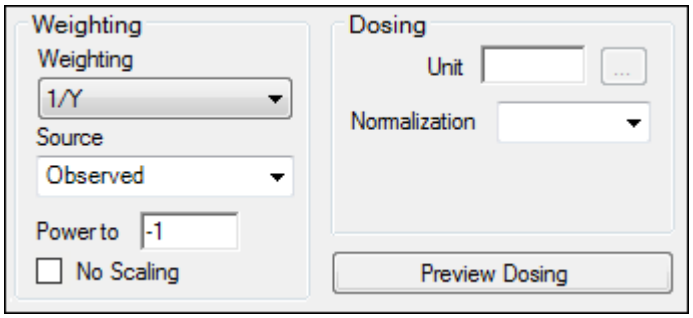
Note: The completed project (`IndivPK.phxproj`) is available for reference in …`\Examples\NLME`.

Set up the Least-Squares Regression PK model object

1. Create a new project called `IndivPK`.
2. Import the datasets `IndivPK11.xls` and `IndivPK11_Dose.xls` from …`\Examples\NLME\Supporting files`.
3. In the *File Import Wizard* dialog, click the **Next Arrow** button to move from the IndivPK11 dataset to IndivPK11 and click **Finish**.
4. Right-click the **IndivPK11** worksheet and select **Send To > Modeling > Least Squares Regression Models > PK Model**.
5. Rename the object as `WNL Model`.
6. Use the option buttons in the Main Mappings panel to map the data types to the following contexts:
   Time to the **Time** context.
   Conc to the **Concentration** context.
7. Click **Dosing** in the Setup tab.
8. Use the mouse pointer to drag the **IndivPK11_Dose** worksheet from the Data folder to the Main Mappings panel. Data types should automatically map as follows:
   Time to the **Time** context.
   Dose to the **Dose** context.
9. In the **Model Selection** tab, scroll down the table of models and check the box for model **11**.

Selected	Number	Input	Compartments	MicroMacro	LagTime	Elim
☐	9	IV-Infusion	2	micro	No	1st or
☐	10	IV-Infusion	2	macro	No	1st or
☑	11	1st Order	2	micro	No	1st or
☐	12	1st Order	2	micro	Yes	1st or

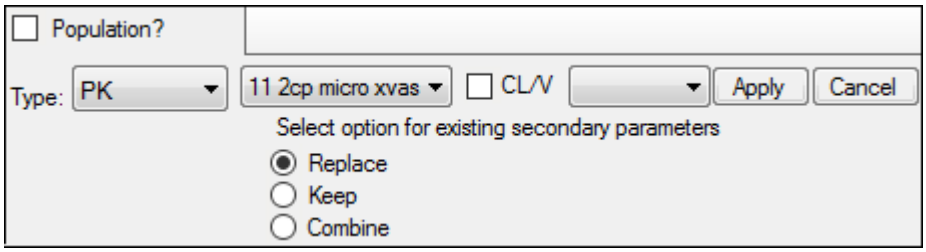
10. Select the **Weighting/Dosing Options** tab.
11. In Weighting menu select **1/Y**.



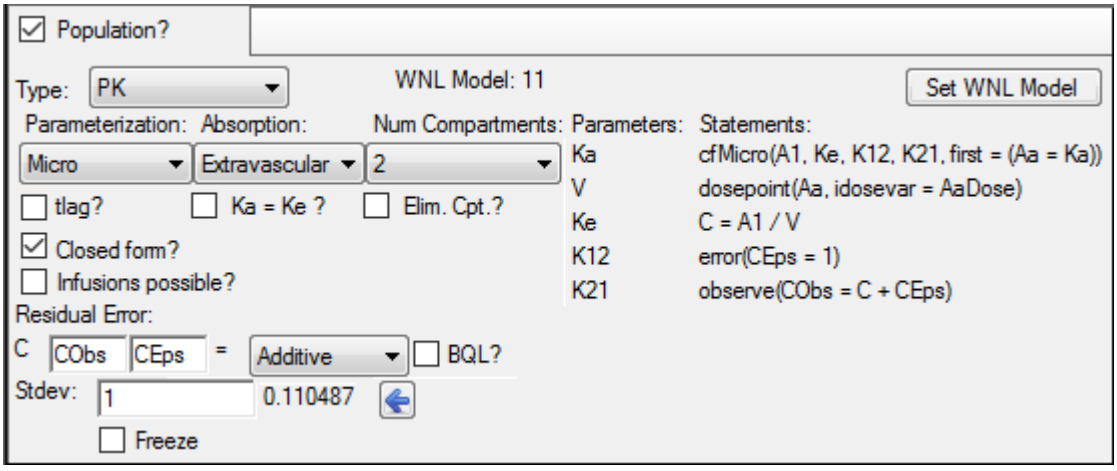
12. Click (**Execute** icon) to execute the object.

Create the Maximum Likelihood model object

1. Right-click the **IndivPK11** worksheet and select **Send To > Modeling > Maximum Likelihood Models**.
2. Rename the object as `NLME Model`.
3. In the Structure tab, uncheck the **Population** box.
4. Click the **Set WNL Model** button.
   The contents of the Structure tab changes. The first of the two untitled menus allows users to select a PK model, and the second allows users to select a PD model.
5. In the first untitled menu, select Model 11 (**11 2cp micro xvas**).



6. Make sure the second untitled menu is clear and click **Apply** to set the model.



7. Use the option buttons in the Main Mappings panel to map the data types to the following contexts:
   Time to the **Time** context.
   Conc to the **CObs** context.
8. Click **Dosing** in the Setup tab.

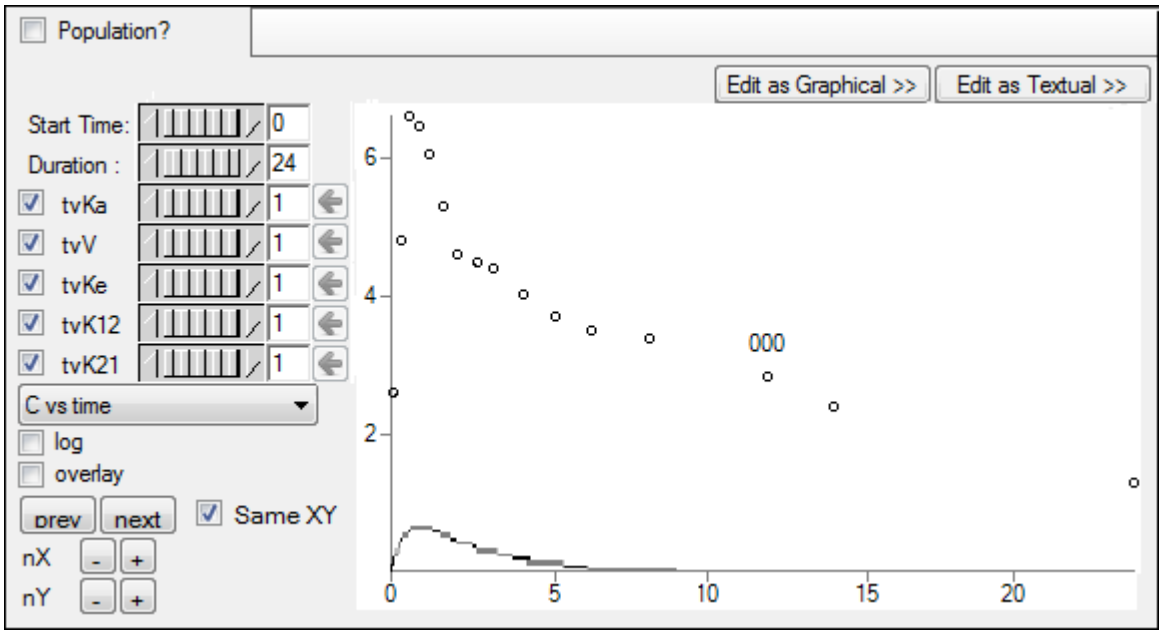
9. Use the mouse pointer to drag the **IndivPK11_Dose** worksheet from the Data folder to the Main Mappings panel.
10. Use the option buttons in the Dosing Mappings panel to map the data types to the following contexts:
    Dose to the **Aa** context.
    Time to the **Time** context.
11. Execute the object.

Initial Estimates configuration

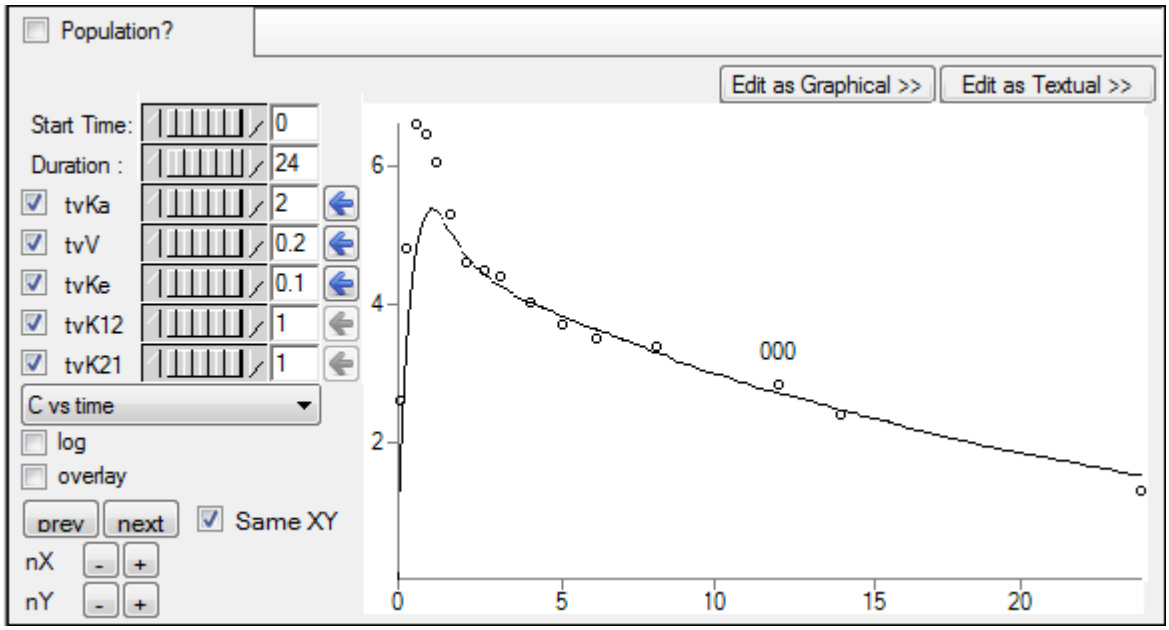
The Initial Estimates tab can be used to find the best initial estimates for different parameters.

1. Select the **Initial Estimates** tab.

   Adjust the portion of the plot to view using the **Start Time** and **Duration** options.
2. For this example, set the **Duration** to 24 either using the slider or typing the value in the field next to the slider.



3. Enter different values for the fixed effect parameters, as shown in the table below, and see how each adjustment affects the results on the XY plot. Use the slider or typing the value in the field next to the slider.
   tvKa = 2
   tvV = 0.2 (make sure to re-check the checkbox)
   tvKe = 0.1 (make sure to re-check the checkbox)
   tvK12 = 1
   tvK21 = 1



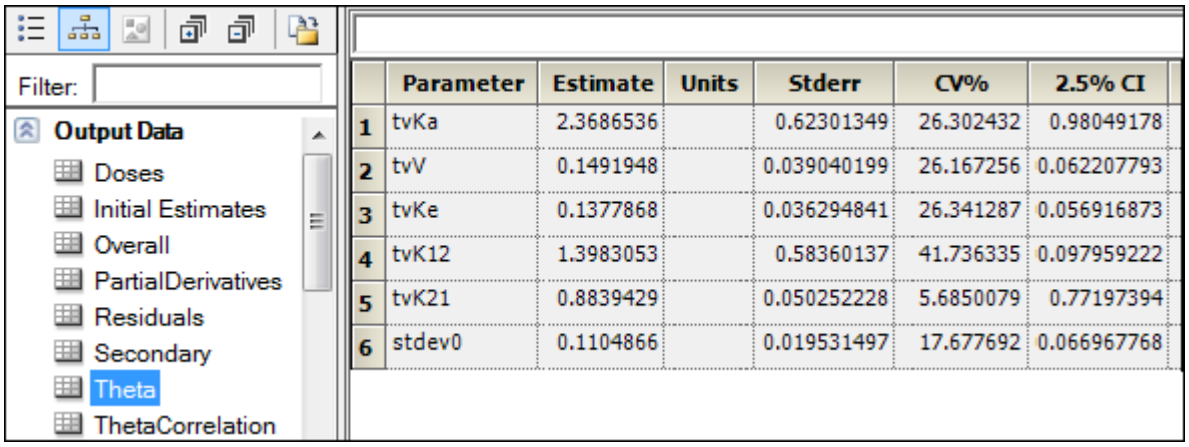
4. Click the left arrow button beside each parameter to copy the new estimate to the Initial estimate field.

   For more information on using the Initial Estimates, see the "Initial Estimates tab" description.

5. Execute the object.

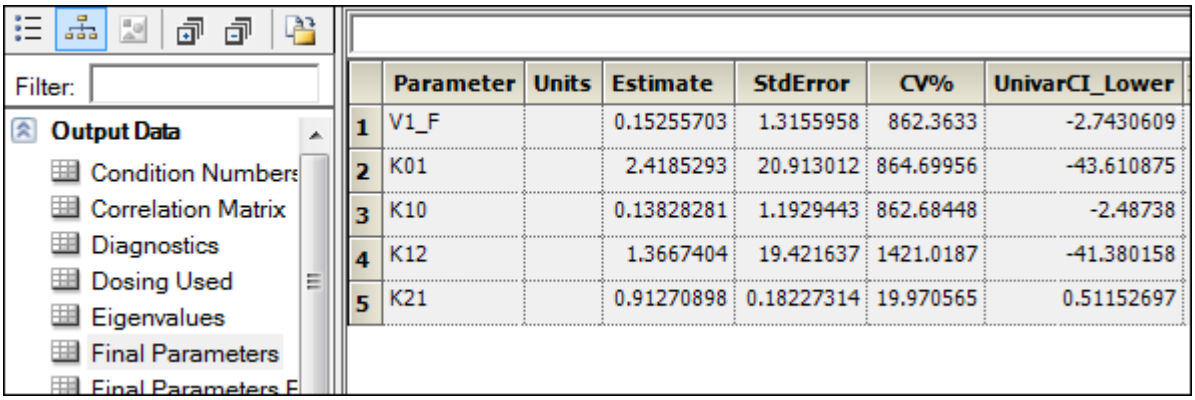
Comparing the results

1. Open the **Theta** worksheet in the Results tab for the Phoenix Model object.



	Parameter	Estimate	Units	Stderr	CV%	2.5% CI
1	tvKa	2.3686536		0.62301349	26.302432	0.98049178
2	tvV	0.1491948		0.039040199	26.167256	0.062207793
3	tvKe	0.1377868		0.036294841	26.341287	0.056916873
4	tvK12	1.3983053		0.58360137	41.736335	0.097959222
5	tvK21	0.8839429		0.050252228	5.6850079	0.77197394
6	stdev0	0.1104866		0.019531497	17.677692	0.066967768

2. Open **Final Parameters** worksheet in the Results tab of the WNL Model object.



	Parameter	Units	Estimate	StdError	CV%	UnivarCI_Lower
1	V1_F		0.15255703	1.3155958	862.3633	-2.7430609
2	K01		2.4185293	20.913012	864.69956	-43.610875
3	K10		0.13828281	1.1929443	862.68448	-2.48738
4	K12		1.3667404	19.421637	1421.0187	-41.380158
5	K21		0.91270898	0.18227314	19.970565	0.51152697

The names of parameters in Maximum Likelihood Models have the following equivalents in the PK Model.

Phoenix_Model	WinNonlin_PK_Model
tvKa	K01
tvV	V1_F
tvKe	K10
tvK12	K12
tvK21	K21

Compare the results of estimated fixed effects. As one can see, the estimates in these models are close but not equal due to different method of model fitting.

This concludes the individual modeling example.

Modeling circadian rhythm, inter-occasion variability, and two epsilons

This example is based on the study of the pharmacokinetics of midazolam reported by Van Rongen, A., et al. (2015) ("Population Pharmacokinetic Model Characterizing 24-Hour Variation in the Pharmacokinetics of Oral and Intravenous Midazolam in Healthy Volunteers". *CPT: Pharmacometrics & System Pharmacology*, 4.8:454–464). The aim of this study was to evaluate 24-hour variation in the pharmacokinetics of the CYP3A substrate midazolam.

This example shows an implementation of the cosinor procedure for describing 24 hour variation. This project also illustrates the visual predictive check results with stratification on inter-occasion variables and how to implement a two dosepoint model with two epsilons, one for oral and the other one for IV using Textual Mode.

Open the project

1. Load the Phoenix project …`\Examples\NLME\Midazolam_Circadian_IOV_TwoEpsilon_Estimation_VPC.phxproj`.

Two datasets are listed in the Data folder. The dataset **midazolam_sim** consists of observations **CObs**, covariate **OCC,** which depends on the time of dosing and includes 6 dosing occasions (OCC=1 (10:00 AM) 2 (02:00 PM), 3 (06:00 PM), 4 (10:00 PM), 5 (02:00 AM), 6 (06:00 AM)), covariate **OCCTime**, representing the time of dosing in minutes starting at midnight, where **TAD** is time after dosing in minutes and is mapped to **Time** to avoid unnecessary computations. There are 12 subjects for each dosing period.

The input doses in the dataset `midazolam_dose` are the same as in the mentioned article. The PO dose separated in time from IV bolus by 150 minutes.

Review the model text

1. In the Object Browser, expand the Workflow and select the **MidazolamModel** object.
2. Select **Model** in the Setup list. The Model text panel is displayed.

The value of bioavailability (F1) oscillates around mesor, and is defined in the structure model parameters as follows:

```
F1=tvF1*exp(nF1+IOV)+ tvAMF*cos((6.283/tvFreq)*(t+OCCTime – tvACF))
```

where:

tvF1 presents the population mean of bioavailability (theta)
nF1 is the Inter-individual variability (eta)
IOV is the interoccasion variability implemented in the following statement with inter-individual variability incorporated inside.

```
IOV=(OCC==1)*nF1_OCC1+(OCC==2)*nF1_OCC2+(OCC==3)*nF1_OCC3+(OCC==4)*nF1_OCC4+(OCC==5)*nF1_OCC5+(OCC==6)*nF1_OCC6
```

tvAMF is the amplitude of cosine function
tvFreq is the period of cosine function (24 hours)
tvACF is the acrophase (time to peak of the cosine function)
t+OCCTime represents the time in minutes starting at midnight.

The clearance is defined similarly as the bioavailability. For the absorption rate constant *Ka*, a multiplication factor at 2:00 PM is added by applying of the ternary operator:

```
Ka=(OCC == 2 ? tvKa*exp(nKa)*tvKaMult : tvKa*exp(nKa))
```

It is worth noting that the multiplication factor in the absorption rate was found in Van Rongen, A., et al. (2015) to give a better fit to the clinical data.

Note that for this example the value of the standard deviation for the residual error changes after TAD reaches 150 minutes; that is, there are two epsilons (epsilon 1: oral, epsilon 2: IV bolus 150 minutes after oral). To incorporate this, a fixed effect is introduced using the following statements:

```
bIV=(t >= 150 ? 1 : 0)
error(CEps= 0.163723)
observe(CObs=Central*(1+(1+bIV*Coeff_IV)*CEps))
```

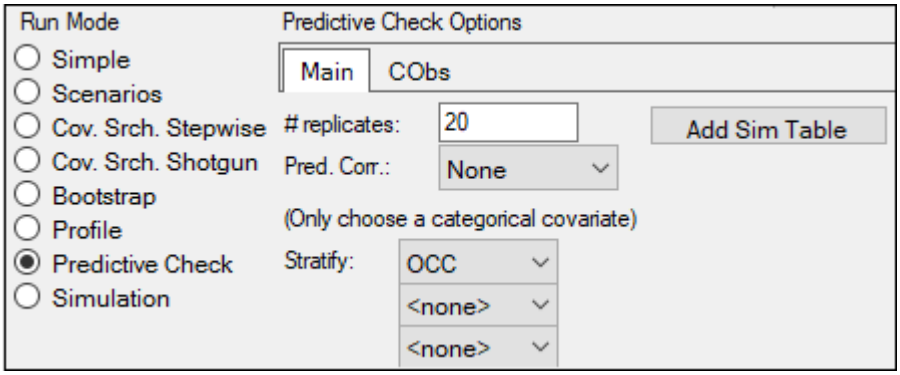
The above statements means that if TAD is less than 150 minutes, the standard deviation for the residual error is 0.163723. Otherwise, it is (1+Coeff_IV)* 0.163723.

Execution of VPC model

1. In the Object Browser select **VPC MidazolamModel** object.

   This model is a copy of the previous one with the values for the fixed and random effects chosen as the final estimated parameter values.

2. Select the **Run Options** tab.
3. Make sure that **Predictive Check** run mode is selected.
4. In the Main sub-tab, leave the categorical covariate **OCC** selected in the **Stratify** menu.



5. Switch to the CObs sub-tab and make sure that the **Quantile** checkbox is checked.
6. Click (**Execute** icon).
7. Select **Pop PredCheck ObsQ_SimQ** in the Plots part of the Results tab.

   For each strata, the observed quantiles are superimposed with predictive check quantiles over the observed data. The results are shown in the image below, which shows a good match between observed and predicted data.

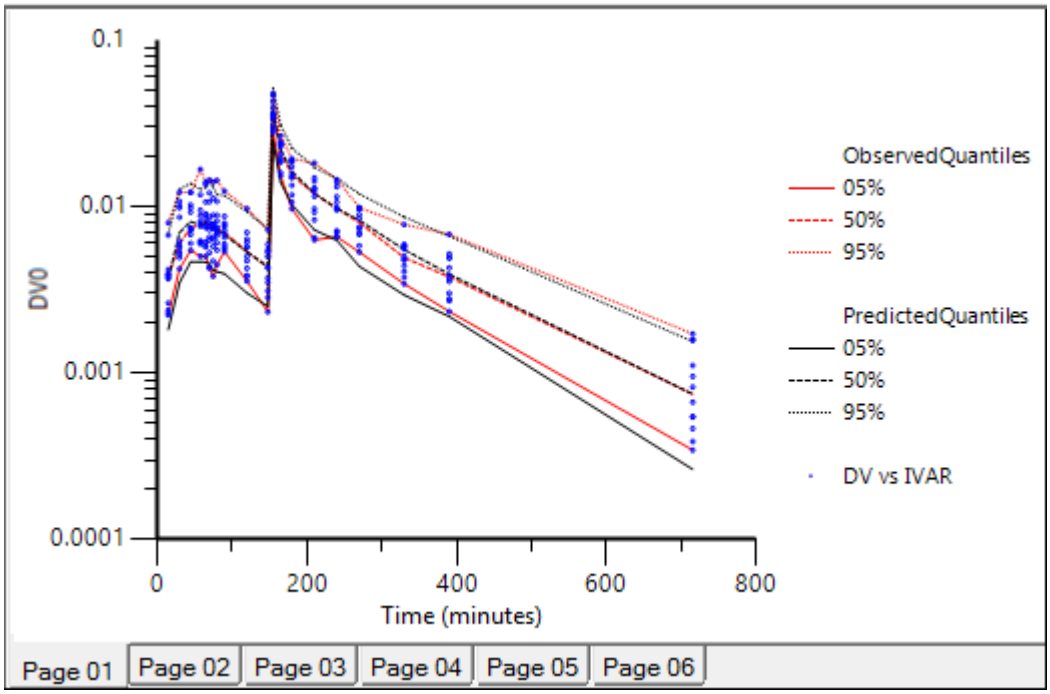


Figure 32-1. Predictive Check plot for VPC MidazolamModel. Red lines: observed quantiles, black lines: predicted quantiles, blue symbols: observed data.

8. Select **Pop PredCheck_ObsQ_SimQCI** in the Plots part of the Results tab.

PredCheck_ObsQ_SimQCI plot of confidence intervals of the simulated quantiles is generated due to checked **Quantile** checkbox in the Predictive Check Options area:

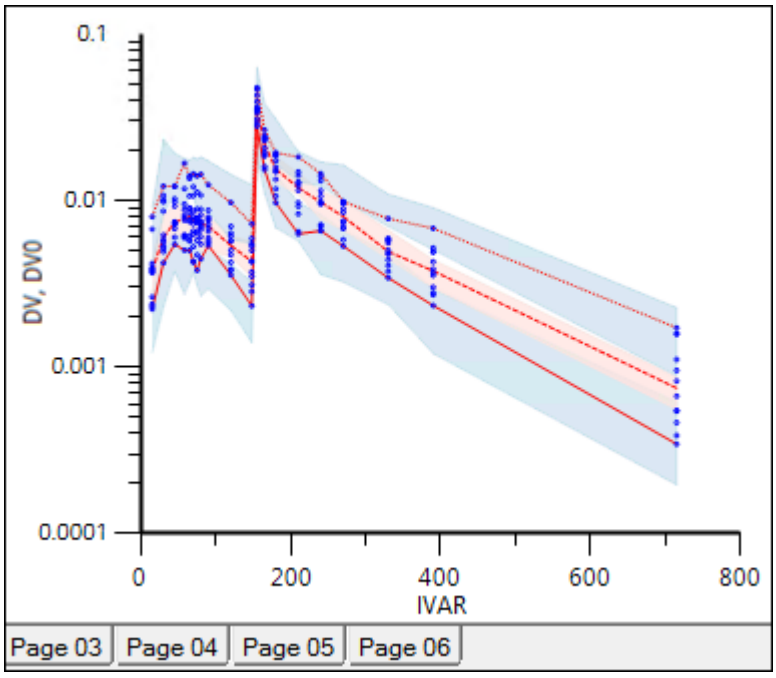


Figure 32-2. Predictive Check plot with confidence intervals of the simulated quantiles for VPC MidazolamModel.

In each plot, the two blue shaded areas respectively represent the confidence intervals for the simulated 5th and 95th quantiles, and the red shaded area represents the confidence interval for the 50th quantile. The red lines inside these shaded areas denote the median of these simulated quantiles.

In this project, the lines that define the outer edge of the shaded areas have been removed by double-clicking on the plot and setting **Line Style** to **None** for each of the Predicted Quantiles graphs that named "05%, 05%" through "95%. 95%".

Maximum Likelihood Model Comparer

The Maximum Likelihood Model Comparer is an operational object that can compare any executed Maximum Likelihood Model in a project, calculate differences between some model diagnostics, as well as calculate p-values for nested models.

Use one of the following to add the object to a Workflow:

Right-click menu for a Workflow object: **New > Modeling > Maximum Likelihood Model Comparer**.
Main menu: **Insert > Modeling > Maximum Likelihood Model Comparer**.
Right-click menu for a worksheet: **Send To > Modeling > Maximum Likelihood Model Comparer**.

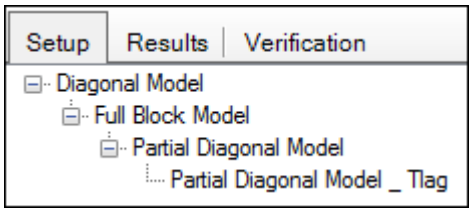
Note: To view the object in its own window, select it in the Object Browser and double-click it or press **ENTER**. All instructions for setting up and execution are the same whether the object is viewed in its own window or in the Phoenix view.

This section contains information on the following topics:

User interface description

Setup tab

The Setup tab lists all the Maximum Likelihood Models objects in a project.



Note: Avoid having multiple Maximum Likelihood Models objects with the same name, even if they are in different workflows. Model Comparer will include all models with the same name in the comparison, even if their **Compare** boxes are not checked (they will become checked at execution). If the names cannot be changed for some reason, be sure to use the **Hide** checkboxes next to the model objects that are not wanted for the comparison.

If a Maximum Likelihood Models object was created by making a copy of a previous Maximum Likelihood Models object, then the copy is nested underneath the first model in the Comparison panel. The original model is the root model, and the copied model is the child.

Model copies that are nested underneath the original model will typically have additional parameters that are compared against the original model, which has fewer parameters.

In the Setup tab users can change the hierarchical relationship between root and sub-models.

- Use the pointer to select a model in the Setup tab and drag the model on top of another model.
- In the dialog, click **Yes** to confirm the move.

  The selected model is nested underneath the selected root model.
- To move a child model back to the root level, right-click the child model and select **Orphan Model**.
- In the dialog, click **Yes** to confirm the move.

  The model is moved back to the root level in the Setup tab.

Note: Although there are no icons in the Setup tab for importing/saving/loading object settings, these operations are still available using the **File > Import** menu, and right-clicking the object in the Object Browser or in the workflow diagram.

The area to the right of the model allows users to hide, remove, and select models to compare.

Hide	Compare	Name	Sort	Method	Descrip	Lineage	LogLik	-2(LL)	A
☐	☐	Diagonal Model		FOCE L-B			-179.334	358.6681	37
☐	☐	Full Block Model		FOCE L-B		Diagonal	-173.3223	346.6445	36
☐	☐	Partial Diagonal Model		FOCE L-B		Full Block	-173.495	346.99	36
☐	☐	Partial Diagonal Model_		FOCE L-B		Partial Di	-151.6901	303.3801	32

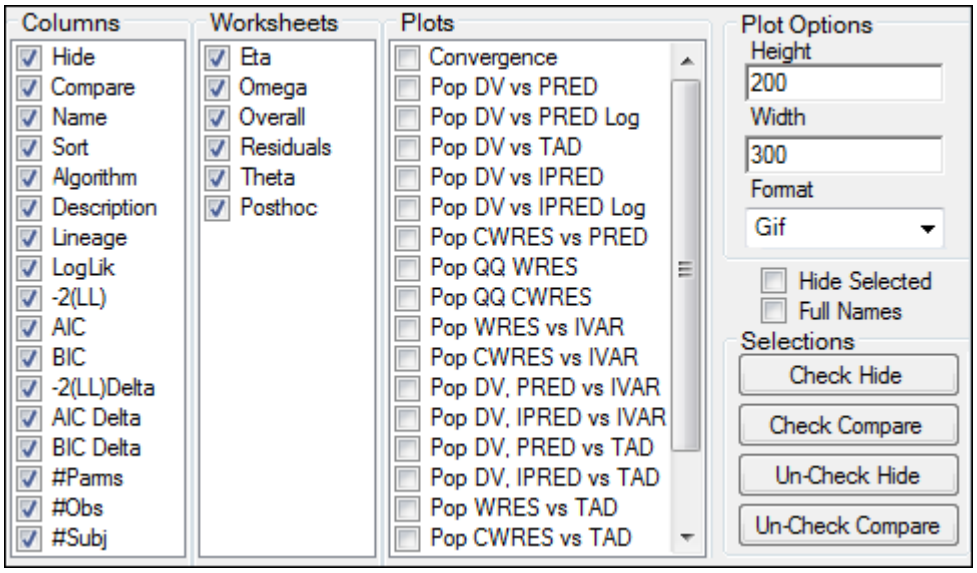
Check or uncheck the **Hide** or **Compare** checkboxes to exclude or include a model in the comparison.

- If the **Hide** checkbox is selected, then the model is considered "hidden" and is removed from any comparisons.
- If the **Compare** checkbox is selected, then the selected model is included in comparisons.

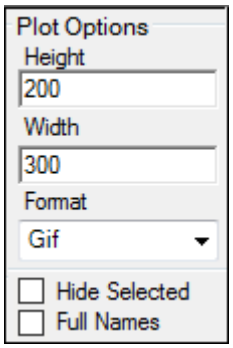
Additional model descriptions can be entered in the **Description** field by clicking the field twice and typing in the field. Description is carried over to the result worksheets.

Options tab

The Options tab lists all the columns, worksheets and plots available for comparisons.

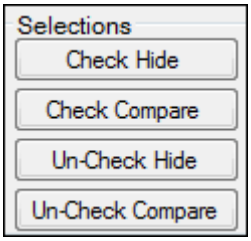


- Uncheck a column checkbox to remove that column from the Comparison panel. Removed columns are not included in the output worksheets.
- Check the checkboxes in the Worksheets and Plots lists to include worksheets and plots in the comparison results.
- Uncheck the checkboxes in the Worksheets and Plots lists to exclude the worksheets and plots from the comparison results.
- In the **Plot Options** area, select the height, width, and graphic format for the output plot.



- If a Maximum Likelihood Model object is marked as hidden, then check the **Hide Selected** checkbox to remove the model from the Setup tab. The hidden model is not used in any comparisons.
- Check the **Full Names** checkbox to display the full name of a model. The full name includes the name of the workflow containing the model.

  Use the buttons in the **Selections** area to select or hide several models at once. Multiple rows in the Comparison panel must be selected to use these buttons.



- To select multiple rows in the Comparison panel, press and hold the CTRL key and right-click beside each row to select it.
  Or
  Use the pointer to select a row. Press and hold the SHIFT key and use the up or down arrow keys to select multiple rows.
- When the desired rows are selected, users can do the following:

  Click **Check Hide** to hide the selected models.
  Click **Check Compare** to include the selected models in the comparison.
  Click **Un-Check Hide** to not hide the models.
  Click **Un-Check Compare** to remove the models from the comparison.

Results

Worksheet

- **Eta**: Empirical Bayesian estimates of random differences from population expected values for each subject and occasion. This worksheet presents eta estimates by scenario and by model being compared.
- **Omega**: Estimates of the random effects variance-covariance matrix, correlation matrix and shrinkage. This worksheet is presented by scenario and by model being compared.
- **Overall**: Stacks each model 'Overall' worksheet result but doesn't present any comparison calculation. Columns include:

  Name: Model names being compared (as displayed in the Object Browser)
  Description: Model description text if entered by the user in the setup tab.
  Scenario: Name of the scenario (if applicable)
  Retcode: Code indicating the status of the run convergence
  LogLik: Loglikelihood
  -2LL: –2 Loglikelihood
  AIC: Akaike Information Criterion for each model run
  BIC: Bayesian Information Criterion for each model run
  nParm: Number of model parameters
  nObs: Number of observations
  nSub: Number of subjects
  EpsShrinkage: Epsilon shrinkage
- **Posthoc**: Compares the Posthoc worksheet estimates of selected models. Model name, any description, scenario, replicate, ID, time, and one column for each structural parameter are listed.
- **Residuals**: This table stacks 'Residuals' worksheet results of each individual model with the addition of Model name and Description. It contains columns for:

  Name: Model name
  Description: Model description
  Scenario: Scenario name if applicable
  ID: Subject ID
  Time: Time
  TAD: Time after dose
  PRED: Population prediction
  IPRED: Individual prediction
  DV: Observed dependent variable. This column header will display the name of the column mapped to CObs for each individual run. If models being compared have different column names mapped to CObs then as many columns as CObs names used will display.
  IRES: Individual residual

PREDSE: Standard error of predicted value
Weight: Applied model weight
IWRES: Individual weighted residual
WRES: Weighted residual
CWRES: Corrected weighted residual
PCWRES: Predictive check weighted residuals
CdfPCWRES: Cumulative distribution function for predictive check weighted residual. This column would only have non-zero values if the PCWRES option was selected in the individual runs.
CdfDV: Cumulative distribution function for the dependent variable. This column would only have non-zero values if the PCWRES option was selected in the individual runs.
TADSeq: Time after dose sequence. This variable is used to produce plots with TAD.
ObsName: Observation Name. This variable allows to distinguish observations when there are multiple observed quantities.
Var. Inf. factor: The variance inflation factor (if applicable)

- **Theta**: This table stacks 'Theta' worksheet result of each individual model run with the addition of Model name and Description. It contains columns for:

  Name: model name
  Description: model description
  Scenario: scenario name if applicable
  Estimate: parameter estimates
  Parameter: parameter name
  Units: parameter units (if applicable)
  Stderr: standard error
  CV%: percent confidence of variation
  lower % CI: lower confidence interval
  upper % CI: upper confidence interval
  Var. Inf. Factor: variance inflation factor (if applicable)

- **Comparison Results**: List all columns checked under options. Columns may include:

  Hide: Models hidden
  Compare: Models to be compared in the plot table
  Method: phoenix model engine used for each model
  Name: is the name of the object being compared
  Description: optional description of the model
  Lineage: List the 'parent' (i.e. reduced model) if the current model is a 'child' (derived from but with additional parameters) of another model
  LogLik: estimate of the loglikelihood upon convergence
  AIC: Akaike Information Criterion for each model run
  BIC: Bayesian Information Criterion for each model run
  -2(LL)Delta: Difference in -2LL for nested models only (Lineage-Child model)
  AICDelta: Difference in AIC values. for nested models only (Lineage-Child model)
  BICDelta: Difference in BIC values for nested models only (Lineage-Child model)
  #params: Number of model parameters
  #obs: Number of observations
  #Subj: Number of subjects
  p-value: Chi-Square p-value based on the Likelihood Ratio Test for nested models only.

Plot

- **Plot Table**: Illustration of all the selected plots in rows for each model being compared (columns). This plot table is not editable and it places side by side the plots from the individual runs for easy visualization. Any changes done to the individual run plots would reflect in this Plot table. This table can be exported or printed upon right click.

Text File

- **Settings**: A text format file with the list of current columns, plots and models being compared.

Index