

# Package: Certara.RsNLME.ModelBuilder (via r-universe)

December 21, 2024

**Title** Pharmacometric Model Building Using 'shiny'

**Version** 3.0.1

**Description** Develop Nonlinear Mixed Effects (NLME) models for pharmacometrics using a 'shiny' interface. The Pharmacometric Modeling Language (PML) code updates in real time given changes to user inputs. Models can be executed using the 'Certara.RsNLME' package. Additional support to generate the underlying 'Certara.RsNLME' code to recreate the corresponding model in R is provided in the user interface.

**Depends** R (>= 4.0)

**License** LGPL-3

**URL** <https://certara.github.io/R-RsNLME-model-builder/>

**Encoding** UTF-8

**RoxygenNote** 7.3.2

**Suggests** knitr, rmarkdown, testthat (>= 3.0.0)

**Imports** Certara.RsNLME, shinymeta, shinyAce, bslib (>= 0.7.0), data.table, DT, ggplot2, ggforce, htmltools, htmlwidgets, magrittr, methods, shiny (>= 1.7.0), shinyjs, shinyWidgets, tools, utils, fs

**Config/testthat/edition** 3

**NeedsCompilation** no

**Author** James Craig [aut, cre], Mike Talley [aut], Shuhua Hu [ctb], Certara USA, Inc [cph, fnd]

**Maintainer** James Craig <james.craig@certara.com>

**Date/Publication** 2024-12-20 11:00:10 UTC

**Config/pak/sysreqs** libfontconfig1-dev libfreetype6-dev git make libicu-dev libssh-dev libssh2-1-dev libxml2-dev libssl-dev zlib1g-dev

**Repository** <https://certara-jcraig.r-universe.dev>

**RemoteUrl** <https://github.com/cran/Certara.RsNLME.ModelBuilder>

**RemoteRef** HEAD

**RemoteSha** 34c1b33f3a34b2547b53390787a42a791e016883

## Contents

create_metamodelBuiltin . . . . .	2
create_metamodelTextual . . . . .	3
estimatesUI . . . . .	3
modelBuilderUI . . . . .	4
modelTextualUI . . . . .	5

<b>Index</b>	<b>7</b>
--------------	----------

---

create\_metamodelBuiltin

*Run modelBuilderUI() and create resulting metamodel*

---

### Description

Used by Pirana to run modelBuilderUI(), saving the resulting metamodel e.g., .mmdl file given model building operations performed in GUI.

### Usage

```
create_metamodelBuiltin(metamodelFile, datafile, author = "")
```

### Arguments

metamodelFile	File where the metamodel should be created.
datafile	File with input data.
author	Optional character string to specify the author in the metamodel.

### Value

NLME PML model S4 class instance

### Examples

```
if (interactive()) {
  tmp_data <- tempfile(fileext = ".csv")
  write.csv(Certara.RsNLME::pkData, tmp_data, row.names = FALSE)

  create_metamodelBuiltin(
    "run1.mmdl",
    tmp_data
  )
}
```

---

`create_metamodelTextual`*Send metamodel to modelTextualUI() and run shiny application*

---

**Description**

Used by Pirana to send existing metamodel to modelTextualUI() for editing, and after returning, saves it as a metamodel file e.g, .mmdl.

**Usage**

```
create_metamodelTextual(metamodelFile)
```

**Arguments**

metamodelFile Path to existing metmodel file.

**Details**

If DOSING CYCLE block is presented in the metamodel, it will be transferred to COLDEF block with a warning.

**Value**

Updated metamodel text.

**Examples**

```
if (interactive()) {  
  mmdl_file <- system.file("vignettesdata/OneCpt_IVInfusion.mmdl",  
    package = "Certara.RsNLME")  
  
  create_metamodelTextual(  
    mmdl_file  
  )  
}
```

---

`estimatesUI`*Shiny GUI to examine the model and evaluate estimates for fixed effects.*

---

**Description**

Shiny GUI to examine the model and evaluate estimates for fixed effects.

**Usage**

```
estimatesUI(model, host = NULL)
```

**Arguments**

model	Model object.
host	Optional host parameter of class hostParams. If NULL, local host will be used.

**Value**

A model object of class NlmePmlModel

**Examples**

```
if (interactive()) {  
  library(Certara.RsNLME)  
  host <- hostParams(  
    parallelMethod = "None",  
    hostName = "local",  
    numCores = 1  
  )  
  
  model <- pkmodel(  
    parameterization = "Clearance",  
    absorption = "Intravenous",  
    numCompartments = 2,  
    data = pkData,  
    ID = "Subject",  
    A1 = "Amount",  
    CObs = "Conc",  
    Time = "Act_Time",  
    modelName = "pk_model"  
  )  
  
  model <- estimatesUI(model, host)  
}
```

---

modelBuilderUI

*Build RsNLME model from Shiny GUI and generate corresponding  
RsNLME code*

---

**Description**

Shiny application to build RsNLME model from Shiny GUI and generate corresponding RsNLME code based on input selections.

**Usage**

```
modelBuilderUI(  
  data,  
  modelName = "PKPDmodel",  
  workingDir = "",  
  baseModel = NULL  
)
```

**Arguments**

data	Input dataset.
modelName	Name of the model; if missing, named as 'PKPDmodel.'
workingDir	Working directory to run the model. Current working directory will be used if workingDir not specified or does not exist.
baseModel	The model object from where the input dataset and model name are recovered if arguments data and modelName are not specified.

**Value**

A model object of class `NlmePmlModel`

**Examples**

```
if (interactive()) {  
  model <- modelBuilderUI(data = Certara.RsNLME::pkData, modelName = "PK_Model")  
}
```

---

modelTextualUI

*Edit textual RsNLME model from Shiny GUI*

---

**Description**

Shiny application to update RsNLME model from Shiny GUI and directly edit PML statements using Ace editor. Syntax and semantic check is performed by TDL executable (if presented). The Shiny application also allows adding input options and column mappings from Shiny GUI.

**Usage**

```
modelTextualUI(baseModel, initpml, data, modelName = "PKPDmodel")
```

**Arguments**

<code>baseModel</code>	The model object from where the information is recovered.
<code>initpml</code>	Initial PML model file to be edited. Overrides <code>baseModel@statements</code> , if presented.
<code>data</code>	Input data frame. Overrides <code>baseModel@inputData</code> , if presented.
<code>modelName</code>	Name of the model; if missing, named as 'PKPDmodel'. Overrides <code>baseModel@modelInfo@modelName</code> , if presented.

**Value**

A model object of class `NlmePmlModel`

**Examples**

```
if (interactive()) {  
  model <- modelBuilderUI(data = Certara.RsNLME::pkData, modelName = "PK_Model")  
  
  model <- modelTextualUI(baseModel = model)  
}
```

# Index

`create_metamodelBuiltin`, 2

`create_metamodelTextual`, 3

`estimatesUI`, 3

`modelBuilderUI`, 4

`modelTextualUI`, 5