

EXECUTE user guide

PsN 4.6.0

Revised 2016-05-05

1 Introduction

The execute script is a PsN tool that allows you to run multiple modelfiles either sequentially or in parallel. It is an nmfe replacement with advanced extra functionality.

Execute creates subdirectories where it puts NONMEMs input and output files, to make sure that parallel NONMEM runs do not interfere with each other. The top directory is by default named 'modelfit_dirX' where 'X' is a number that starts at 1 and is increased by one each time you run execute. The name of the directory can be changed with options -directory, -model_dir_name or -timestamp, see list of options below. Example

```
execute -threads=2 -retries=5 run1.mod
```

2 Output

When the NONMEM runs are finished, output and table files will be copied to the directory where execute started in which means that you can normally ignore the run directory. If you need to access any special files you can find them inside the run directory. The run directory also contains a raw_results csv-file, a comma-separated file with summary information about minimization status, run time, ofv, parameter estimates etc. In the run directory you find a few subdirectories named 'NM_runY'. For each model file specified on the command line there will be one 'NM_runY' directory in which the actual

NONMEM execution takes place. The order of the 'NM_runY' directories corresponds to the order of the modelfiles given on the command line. The first run will take place inside 'NM_run1', the second in 'NM_run2' etc.

3 Required input

At least one model file is required on the command-line.

4 Optional input

All options listed in document `common_options.pdf` apply to execute. Those options govern how the NONMEM runs are managed, and hence apply to all PsN scripts that run NONMEM. A large selection of those options are included in this document, together with the options that only apply to execute.

4.1 Basic options

The following options are the ones most commonly used.

-h or -?

Print the list of available options and exit.

-help

With `-help` all programs will print a longer help message. If an option name is given as argument, help will be printed for this option. If no option is specified, help text for all options will be printed.

-directory = 'string'

Default `modelfit_dirN`, where N will start at 1 and be increased by one each time you run the script. The `directory` option sets the directory in which PsN will run NONMEM and where PsN-generated output files will be stored. You do not have to create the directory, it will be done for you. If you set `-directory` to a the name of a directory that already exists, PsN will run in the existing directory.

-seed = *'string'*

You can set your own random seed to make PsN runs reproducible. The random seed is a string, so both `-seed=12345` and `-seed=JustinBieber` are valid. It is important to know that because of the way the Perl pseudo-random number generator works, for two similar string seeds the random sequences may be identical. This is the case e.g. with the two different seeds 123 and 122. Setting the same seed guarantees the same sequence, but setting two slightly different seeds does not guarantee two different random sequences, that must be verified.

-clean = *'integer'*

Default 1. The clean option can take four different values:

0 Nothing is removed

1 NONMEM binary and intermediate files except INTER are removed, and files specified with option `-extra_files`.

2 model and output files generated by PsN restarts are removed, and data files in the `NM_run` directory, and (if option `-nmqual` is used) the xml-formatted NONMEM output.

3 All `NM_run` directories are completely removed. If the PsN tool has created `modelfit_dir:s` inside the main run directory, these will also be removed.

-nm_version = *'string'*

Default is 'default'. If you have more than one NONMEM version installed you can use option `-nm_version` to choose which one to use, as long as it is defined in the `[nm_versions]` section in `psn.conf`, see `psn_configuration.pdf` for details. You can check which versions are defined, without opening `psn.conf`, using the command

```
psn -nm_versions
```

-threads = *'integer'*

Default 5 (if default PsN config file is used). Use the threads

option to enable parallel execution of multiple models. This option decides how many models PsN will run at the same time, and it is completely independent of whether the individual models are run with serial NONMEM or parallel NONMEM. If you want to run a single model in parallel you must use options `-parafile` and `-nodes`. On a desktop computer it is recommended to not set `-threads` higher the number of CPUs in the system plus one. You can specify more threads, but it will probably not increase the performance. If you are running on a computer cluster, you should consult your system administrator to find out how many threads you can specify.

-version

Prints the PsN version number of the tool, and then exit.

-model_dir_name

Default not used. This option changes the basename of the run directory from `modelfit_dir` to `<modelfile>.dir`. where `<modelfile>` is the name of the (first) input model file, without the extension. This option is specific to `execute`. The directories will be numbered starting from 1, increasing the number each time `execute` is run with a model file with the same name. If the option `directory` is used it will override `-model_dir_name`.

-timestamp

Default not used. This option changes the name of the run directory to `<modelfile>-PsN-<date>` where `<modelfile>` is the name of the first model file in the list given as arguments, without the extension, and `<date>` is the time and date the run was started. Example: directory name `run1-PsN-2014-06-12-152502` for a run that was started at 15:25:02 June 12th in year 2014. This option is specific to `execute`. If the option `directory` is used it will override `-timestamp`.

-copy_data

Default set. Disable with `-no-copy_data`. By default, PsN will copy the datafile into `NM_run1` and set a local path in `psn.mod`, the actual `modelfile` run with NONMEM. If `-no-copy_data` is set, PsN will not copy the data to `NM_run1` and instead set a global path to the datafile in `psn.mod`.

4.2 Auto-generated R-plots from PsN

PsN can automatically generate R plots to visualize results for execute, using a default template found in the R-scripts subdirectory of the installation directory. The user can also create a custom template, see more details in section “Auto-generated R-plots from PsN” in `common_options.pdf`.

The default execute template requires the `xpose4` R library. It also relies on Xpose-type tables being created, such as `sdtab`, `patab` and `cotab`, with the correct run number. See the Xpose documentation for requirements on such table files, e.g. pages 106-108 in

http://xpose.sourceforge.net/bestiarium_v1.0.pdf.

If option `-subset_variable_rplots` is used, the user must ensure that the subset variable is printed to one of the xpose tables, for example `sdtab`, and then there will be separate plots created for subsets of the data, via xpose options `'subset'` and `'by'`. Xpose will treat the subset variable as continuous or categorical based the number of unique values. The template contains a comment line with the xpose command for changing the default classification.

If the conditions for creating the plots are not fulfilled then no pdf will be generated, see the `.Rout` file in the main run directory for error messages.

-rplots = *level*

-rplots<0 means R script is not generated

-rplots=0 (default) means R script is generated but not run

-rplots=1 means basic plots are generated

-rplots=2 means basic and extended plots are generated

Troubleshooting

If no pdf was generated even if a template file is available and the appropriate options were set, check the `.Rout`-file in the main run directory for error messages. If no `.Rout`-file exists then check that R is properly installed, and that either command `'R'` is available or that R is configured in `psn.conf`.

Basic plots

A pdf-file with basic execute rplots will be generated if option `-rplots` is set >0 , and the general rplots conditions fulfilled, see above. The basic plots are created with the `xpose4` functions `basic.gof`, `ranpar.hist`, `ranpar.qq` and a few

others. Please refer to the xpose4 documentation for details regarding these plots.

Extended plots

Extended execute rplots will be generated if option -rplots is set >1 . These are individual plots of ten randomly chosen IDs.

4.3 Retries

A NONMEM run is not always successful and sometimes it is possible to improve the results of the run. A simple procedure is to tweak the initial estimates of the parameters and see if changed starting condition gives better results. The -tweak_inits and -retries options turn on this feature in PsN, which is off by default. When the feature is on and the minimization fails, PsN will pick a random value within 10% above or below each initial value, and run the model again with these new initial estimates. This is called a retry. PsN can do several retries, and each time the bounds is increased by 10%. Note that PsN always respects the upper and lower bound set in the model file.

The retries functionality is adapted to control streams with a single \$PROBLEM. If there are more than one \$PROBLEM, PsN will locate the *first \$PROBLEM that runs estimation in its last \$EST*, i.e. the NONMEM lst-file has the line

```
ESTIMATION STEP OMITTED:                NO
```

for the last \$EST in that \$PROBLEM. PsN will use the termination status and ofv of that estimation to decide if a retry should be initiated and also for selecting the best try. Subsequent \$PROBLEMs that also have \$EST will be ignored in the retries procedure. If no \$PROBLEM runs estimation in the last \$EST, PsN will instead use the *first \$PROBLEM that runs evaluation in its last \$EST*, i.e. the NONMEM lst-file has the line

```
ESTIMATION STEP OMITTED:                YES
```

for the last \$EST in that \$PROBLEM. In this case retries will only be initiated if forced using option -min_retries, see below, and the ofv from the evaluation will be used to select the best retry.

If there are multiple \$PROBLEM and PsN concludes that the initial estimates should be tweaked, PsN will first try with the \$PROBLEM used for decision-making, but if no parameters were found in that \$PROBLEM, e.g. if \$MSFI was used, PsN will try with the previous \$PROBLEM(s), one by one, until either PsN finds parameters to tweak, or there are no more \$PROBLEMS to try. No retry will be initiated if PsN cannot find any parameters to tweak.

The control files and outputs for the retries are found in the NM_run1 subdirectory. The files psn-1.mod and psn-1.lst are for the first run which is always performed and not called a retry. psn-2.mod and up are the control files with perturbed initial estimates. In the same directory as psn-1.mod, psn-2.mod etc are created, there is a file called stats-runs.csv. In there is a set of parameters+values for the set of runs, in the order given by the modelfile numbers. As the last line is written "Selected ..." where it says which model was judged as the best of all the retries.

The lst-file for the selected model is then copied to psn.lst (no number) in the same directory, and also copied back up to the working directory. Same principle for table files. The file psn-x.mod for the selected model is copied to psn.mod (for PsN version 3.x.x and up), but psn.mod is not copied back up.

The string MINIMIZATION SUCCESSFUL is important when PsN decides whether to make a retry. With new estimation methods in NONMEM7, that string will not appear. The flag for minimization_successful is set or unset using the following logic:

1. Only status of last \$EST step is considered, except when last \$EST is IMP with EONLY=1
2. BURN-IN/(REDUCED) STATISTICAL PORTION/OPTIMIZATION NOT TESTED - set
3. BURN-IN/(REDUCED) STATISTICAL PORTION/OPTIMIZATION COMPLETED - set
4. BURN-IN/(REDUCED) STATISTICAL PORTION/OPTIMIZATION NOT COMPLETED PRIOR TO USER INTERRUPT - set
5. BURN-IN/(REDUCED) STATISTICAL PORTION/OPTIMIZATION NOT COMPLETED - unset

6. If any of the two steps in SAEM failed - unset
7. If last \$EST is IMP with EONLY=1, the minimization status is determined by the next to last \$EST

4.3.1 Controlling retries using PsN options

The retries option, which defaults to 0, controls the maximum number of retries before giving up. Note that it is possible to set a different default in the psn.conf, the PsN configuration file. The option -min_retries, with default 0, controls the minimum number of retries that PsN is forced to make, and has precedence over -retries. Option -min_retries is useful if the user suspects that there is a risk of finding local minima.

Normally PsN will be satisfied with a run that has minimization successful, but PsN can be more picky about the quality of NONMEM results. If the -picky option is used PsN will do a retry even if the minimization is successful, given that a 'picky-message' appears in NONMEM's minimization message, see the list of picky-messages below by the picky option description.

If option -reduced_model_ofv is set to some value (option currently only available in execute), PsN will do a retry if the current ofv is more than 'accepted_ofv_difference' worse than reduced_model_ofv.

It is also possible to reduce PsN's requirements of quality. If the minimization is not successful but the significant digits are high enough, PsN can skip doing a retry. The limit is set with option -significant_digits_accept. This option is by default not used, and it has no effect if picky is set at the same time.

Following is the list of conditions at the end of a NONMEM run that will lead PsN to initiate a retry, provided that the maximum number of retries set with option -retries has not been reached:

1. The minimum number of retries set with -min_retries have not yet been run.
2. Option -reduced_model_ofv is set and the run ofv is larger than reduced_model_ofv+accepted_ofv_difference
3. Option -picky is set and the run has either not finished with MINIMIZATION SUCCESSFUL, or finished with one of the picky-messages listed below.

4. The run has not finished with MINIMIZATION SUCCESSFUL and option `-significant_digits_accept` is either not set or the number of significant digits is less than `-significant_digits_accept`.
5. The run has finished fulfilling conditions from MINIMIZATION SUCCESSFUL/`significant_digits/picky/reduced_model_ofv`, but the ofv of this run minus `'accepted_ofv_difference'` is larger than the ofv of a previous run that did not fulfill the conditions, indicating that the current run has terminated in a local minimum.

After all retries have finished, PsN will select the best try. The rules for selection are significantly changed in PsN 4.3.9 and later, giving the ofv value more weight than in earlier versions. In most cases the run with the lowest ofv is selected, but there is a `'accepted_ofv_difference'` preference for runs that pass the picky test, if option `picky` was set, or for minimization successful. The selection procedure is

1. If option `-picky` was set, and the ofv of the best try that passed the picky test is not more than `accepted_ofv_difference` worse than the best overall ofv, then choose the best try that passed the picky test.
2. Otherwise, if the ofv of the best try that had minimization successful is not more than `accepted_ofv_difference` worse than the best overall ofv, then choose the best try that had minimization successful.
3. Otherwise, chose the try with the best overall ofv, if any tries at all gave an ofv.
4. Otherwise, chose the first try.

-tweak_inits

Default set, can be disabled with `-no-tweak_inits`. If NONMEM terminates nonsuccessfully, PsN can perturb the initial estimates and run NONMEM again. The generation of new initial estimates `init_i` for the `i`:th retry are performed according to $init_i = init_0 + rand_uniform(+/-degree*init_0)$ where `init_0` are the initial estimates of the original run and `degree` is set with option `-degree`, see below. The updating procedure makes sure that boundary conditions on the parameters are respected. For this option to have effect, the `-retries` option must be set to a number larger than zero.

- degree** = *'number'*
 Default 0.1. A number larger than 0. This number decides how big the perturbation will be in `tweak_inits`.
- retries** = *'integer'*
 Default 0. The `-retries` option tells PsN how many times it shall try to rerun a NONMEM job if it fails according to given criterias. The `-retries` option is only valid together with `-tweak_inits`.
- min_retries** = *'integer'*
 Default 0. Option `min_retries` forces PsN to try make extra retries even if a run has already terminated successfully, or if estimation is not run at all (e.g. `MAXEVAL=0`). Precedence over `-retries` option.
- picky**
 Default not used. The `-picky` option is only valid together with `-tweak_inits`. Normally PsN only tries new initial estimates if 'MINIMIZATION SUCCESSFUL' is not found in the NONMEM output file. With the `-picky` option, PsN will regard any of the following messages, the 'picky-messages', as a signal for rerunning:
- ```
OESTIMATE OF THETA IS NEAR THE BOUNDARY
OPARAMETER ESTIMATE IS NEAR ITS BOUNDARY
OR MATRIX ALGORITHMICALLY SINGULAR
OS MATRIX ALGORITHMICALLY SINGULAR
```
- significant\_digits\_accept** = *'number'*  
 Default not used. Has no effect in combination with `-picky`. The `-significant_digits_accept` option is only valid together with option `-tweak_inits`. Normally PsN tries new initial estimates if 'MINIMIZATION SUCCESSFUL' is not found in the NONMEM output file. With the `-significant_digits_accept`, PsN will only rerun if the resulting significant digits is lower than the value specified with this option.

**-accepted\_ofv\_difference** = *'number'*

Default 0.5. This option is used by PsN when deciding if a retry should be run, and when selecting the best retry out of the whole set. This option decides how much preference should be given to runs that fulfill the picky conditions/have minimization successful but a slightly higher ofv (at most accepted\_ofv\_difference) than a run that did not fulfill the conditions.

**-reduced\_model\_ofv** = *'number'*

This option is used by PsN when deciding if a retry should be run. It is currently only available with the execute program, so it is strictly speaking not a common option. A retry will be initiated if the ofv of the current run is more than 'accepted\_ofv\_difference' worse than the ofv given with this option.

**-add\_retries**

Default not set. By default, PsN will never do retries on a model when a run is restarted if the file stats-runs.csv is found in the NM\_run subdirectory, since the existence of this file indicates that all retries have finished and the best try has already been selected. If option -add\_retries is set, PsN will ignore that stats-runs.csv exists, and check again if retries are needed based on the existing files in the NM\_run directory. This makes it possible to restart a run using new settings for retries (e.g. -retries, -min\_retries, -picky) compared to the original run. If the original run had clean ≤ 1, PsN will count the previously run tries when relating the number of tries to options -min\_retries and -retries. If clean=2, PsN will not count the previously run tries, but psn.mod which in reality is the finally selected try from the original run will be counted as the first try in the restart.

## 4.4 Output handling and processing

**-nm\_output** = *comma-separated list of file extensions*

Default not used. NONMEM generates many output files per

run. PsN will always copy the lst-file back to the calling directory. The option `-nm_output` decides which of the more than 10 additional files should also be copied back. The default in the auto-generated `psn.conf` is `ext,cov,cor,coi,phi`. Note that NM output files which are not copied to the calling directory can still be found inside the run directory unless `-clean` is set larger than 2. If the user runs `NMQual8` (option `-nmqual` is set) and wants the `log.xml` file copied back then the `log.xml` extension should be included in the `-nm_output` list, i.e. `nm_output=ext,cov,cor,coi,phi,log.xml` or similar.

**`-extra_output`** = *comma-separated list of filenames*

Default not used. If `NONMEM` generates a file which PsN normally does not copy back to the working directory, specifying a comma-separated list of such files with this options will make PsN copy the listed files. An example is output generated by verbatim code.

**`-zip`**

Default not set. If this option is set the `m1` folder in the run directory will be zipped into the file `m1.zip`. This will save space and reduce the number of files generated by PsN. PsN will, also when option `-zip` is not set, automatically unzip zipped `m1` folders when restarting/resuming a run using the `-directory` option.

**`-cwres`**

Default not used. Compute the conditional weighted residuals (CWRES) for a model run. It is also possible for users of `NONMEM7` to request CWRES directly in `$TABLE`.

**`-iofv`**

Default not used. Compute the individual contributions to the objective function (written in file `iotab<N>` in `NM_run` directory). This option is disabled for `NONMEM7` because the additional output phi-file contains individual ofv values.

**`-shrinkage`**

Default not used. Calculate the shrinkage for the model run.

Shrinkage is calculated as  $1 - (\text{sd}(\text{eta}(x)) / \text{omega}(x))$  and measures the shrinkage of the empirical Bayes estimates (EBEs) towards the mean of the expected distribution. A 'large' shrinkage means that diagnostics using EBEs cannot be trusted. The shrinkage values appear in the file `raw_results.csv`. PsN does not use the NONMEM7 shrinkage values. When shrinkage is requested, PsN adds two `$TABLE` to the model file so that NONMEM will output data needed for the shrinkage computation. For eta shrinkage the table requests items ID ETA1 ETA2... and for iwres shrinkage it requests items ID IWRES EVID.

**-mirror\_plots = 'integer'**

Default 0. This command creates a set of simulations from a model file that can then be read into Xpose 4 for mirror plotting. The command requires an integer value `-mirror_plots=XX` where XX is an integer representing the number of simulations to perform. This command uses the MSFO file created by `runN.mod` to get final estimates used in the simulations. If this file is not available `run1.mod` is run again. If run times are long, and you did not create an MSFO file with your initial NONMEM run, you can combine the above command with the `-mirror_from_lst` option to avoid running the model again (PsN then reads from the `*.lst` file to get final parameter estimates for the simulations).

**-mirror\_from\_lst**

Default not used. Can only be used in combination with `-mirror_plots=XX` where XX is an integer representing the number of simulations to perform. These commands create a set of simulations from a model file and output file that can then be read into Xpose 4 for mirror plotting. The `-mirror_from_lst` option reads from the `*.lst` file of a NONMEM run to get final parameter estimates for the simulations.

**-prepend\_model\_file\_name**

Default not used. Table files have the name defined by FILE in `$TABLE`. This option will prepend the model file name, without extension, to the name set with FILE.

#### **-standardised\_output**

Default not used. If this option is used a DDMoRe standardised output (SO 0.2) xml file will be created for each run. Note that the file format is still under development and that this option should be considered experimental. See the `nmoutput2so_userguide.pdf` for more information.

#### **-prepend\_options\_to\_lst**

Default not used. This option makes PsN prepend the final lst-file (which is copied back to the directory from which execute was called) with the file `version_and_option_info.txt` which contains run information, including all actual values of optional PsN options. PsN can still parse the lst-file with the options prepended, so the file can still be used it as input to e.g. `sumo`, `vpc` or `update_inits`. Disabled with `-no-prepend_options_to_lst` if set in `psn.conf`.

## **4.5 Running the nmfe script**

PsN always runs NONMEM via the `nmfe` script (or `NMQual`). The following options govern input for `nmfe`, allowing the user to set options, such as `-parafile`, on the PsN commandline that are passed on unchanged to the `nmfe` script. Please refer to the NONMEM documentation for the meaning and appropriate use of the `nmfe` script options.

#### **-nmfe\_options = *'string'***

Only relevant if NONMEM7.2 or later is used. The text set with this option will be copied verbatim to the `nmfe` script call. PsN will not check that the options are allowed. When set on the PsN commandline the string must be enclosed by quotes if it contains any spaces, but when set in `psn.conf` it must never be enclosed by quotes even if it contains spaces.

On unix type systems, but not on Windows, any parentheses in the `nmfe` options, for example as in `-maxlim=(1,2,3)`, must be escaped with backslashes like this: `-maxlim=\(1,2,3\)`

#### **-parafile = *'filename'***

NONMEM 7.2 (or later version) `parafile`. Appends “`-parafile=filename`” to the `nmfe` call.

**-nodes** = *'integer'*

Only relevant together with option -parafire. Appends “[nodes]=option\_value” to the nmfe call. This option acts independently of -threads. There is no adjustment of -nodes based on -threads or vice versa, and if -threads=1 it is still possible to use -nodes=10.

**-extra\_files** = *comma-separated list of filenames*

Default not used. If you need extra files in the directory where NONMEM is run you specify them in the comma separated -extra\_files list. It could for example be fortran subroutines you need compiled with NONMEM, or a file with initial estimates for the NONMEM7 CHAIN command, a defaults.pnm file (NONMEM 7.2 or later), or a file set in \$ETAS with option FILE.

**-nmfe**

Default set. Invoke NONMEM via the nmfe script (or a custom wrapper) from within PsN. Unless option -nmqual is set, option -nmfe is set automatically. Also, -nmfe is set in the default configuration file.

**-nmqual**

Default not used. Run an NMQual-installed NONMEM via autolog.pl. Only NMQual8 is supported. When set, PsN will locate the autolog.pl file and log.xml in the nmqual subdirectory of the NONMEM installation directory, and then run  
perl /path/autolog.pl /path/log.xml run ce /full/path/workdir psn  
(extra NM options)

Important note: The extra NM options are options set with e.g. -nmfe\_options or -nodes, but unless the do-on-run block of the log.xml file is edited to use these extra options, *they will be ignored*. PsN will append them to the autolog.pl call but it is up to log.xml to decide what to do with them.

If the user wants by default to run PsN with the autolog script it is recommended to set nmqual=1 in the [default\_options] section in psn.conf and remove nmfe=1 from the same section. The user should also consider to add log.xml to the -nm\_output list in psn.conf (see help text for -nm\_output).

## 4.6 Run progress messages

The following options control the messages printed to screen during the run. They do not affect the final results in any way.

### **-display\_iterations**

Default not used. This option turns on display the iterations output from NONMEM during the model run. If the option is not set, the iterations output will be redirected to a file (nmfe\_output.txt). As with any option the user can choose to change the default by editing psn.conf, see the document psn\_configuration.pdf. The template psn.conf distributed with the PsN installation package has this option set as the default for execute, but no other scripts. The option can be disabled with -no-display\_iterations.

### **-silent**

Default not used. The silent option redirects all messages that PsN normally prints to screen to the file run\_messages.txt. Note that messages from NONMEM, such as iterations, will not be printed to run\_messages.txt. Those messages will be in nmfe\_output.txt in the NM\_run subdirectories. Other results and log files are written to disk as usual. Nothing is printed to screen.

### **-verbose**

Default not used. With verbose used, PsN will print more details to screen about NONMEM runs. More precisely PsN will print the minimization message for each successful run and a R:X for each retry PsN makes of a failed run, where X is the run number.

### **-warn\_with\_trace**

Default not used. If -warn\_with\_trace is set, PsN will print a stack trace for all error and warning messages. This is only for developers.

### **-stop\_motion**

Default not used. Used for debugging. Will cause PsN to

pause its execution at certain predefined breakpoints, and only continue after the user hits enter.

**-debug** = *integer*

Default 0. The -debug option is mainly intended for developers who wish to debug PsN. You can set it to '1' to enable warning messages. If you run into problems that require support, you may have to set this and send the output to the developers.

**-tail\_output**

Default not used. This option only works for execute under Windows. Option -tail\_output specifies that execute should invoke a program (tail) that displays the output file, including the gradients, during minimization. The tail program is started automatically but it is up to the user to terminate the program. For the tail\_output option to work, a third party tail program must be installed. Tail programs that are known to work are WinTail and Tail for Win32. The latter is recommended and can be downloaded at <http://tailforwin32.sourceforge.net>. It is also necessary to have correct settings of the variables wintail\_exe, which is the path to the tail program, and wintail\_command, which is the command for the tail program. The defaults, which work for the Tail for Win32 package, are seen below. These two variables can be set to other values in psn.conf.

**-wintail\_exe** = *string*

Default C:\Program Files\Tail-4.2.12\Tail.exe. Only for Windows. See tail\_output for description.

**-wintail\_command** = *string*

Default "tail OUTPUT". Only for Windows. See -tail\_output for description.

## 4.7 Control stream manipulation

**-omega\_before\_pk**

Default not set. In PsN version 3.4.4 and earlier, \$OMEGA was always printed before \$PK. The new default is to keep the

record order of the input model file. To use the old print order, set option `-omega_before_pk`

**-psn\_record\_order**

Default not set. In PsN 4.3.3 and earlier, PsN used an internal record order. The new default is to keep the record order of the input model file. Set this option to use the old order.

**-sde**

Default not set. In PsN version 3.4.4 and earlier, this option made PsN print the records in a particular order suitable for SDE models. The new default is to keep the record order of the input model file. To use the old SDE print order, set option `-sde`.

**-d2u**

Default not set. If set then run `dos2unix` on model files and regular data files.

**-last\_est\_complete**

Default not set. Only applies for models with multiple \$ESTIMATION (NONMEM7 only). Only affects `vpc`, `cdd` if option `-xv` is set, and execute if option `-mirror_plots` or `-predict_model` is set. Indicates that no options needed for the last \$EST are carried over from previous \$EST, all options are set explicitly in that record. See `PsN_and_NONMEM7.pdf` for details.

**-niter\_eonly = *'integer'***

Default undefined. Only applies for NONMEM7 and if last \$EST is IMP or IMPMAP. Only affects `vpc`, `cdd` if option `-xv` is set, and execute if option `-mirror_plots` or `-predict_model` is set. User-chosen value of NITER when estimation is turned off by setting `EONLY=1`. See `PsN_and_NONMEM7.pdf` for details.

#### 4.7.1 Transform Both Sides

PsN can automatically transform a model according to the Transform Both Sides method. For details on the theory read Oberg and Davidian:  
<http://onlinelibrary.wiley.com/doi/10.1111/j.0006-341X.2000.00065.x/abstract>.

Bill Frame: <http://www.ncbi.nlm.nih.gov/pubmed/19904583> <http://www.thtxinfo.com/>

The model must be coded “the Uppsala way”, see below. Also, the user must use untransformed data (e.g. no log-transformation or similar) in the model, and the possible range of IPRED and DV must not include negative values. If IPRED or DV becomes negative there will be a NONMEM error when running the tbs-modified model.

The "Uppsala way" of encoding residual error is to express the standard deviation of the residual error as a parameter W and setting SIGMA 1 FIX. Examples:

#### **Additive error**

```
W = THETA(5)
Y = IPRED+ERR(1)*W
IRES = DV-IPRED
IWRES = IRES/W
```

#### **Proportional error**

```
W = THETA(5)*IPRED
Y = IPRED+ERR(1)*W
IRES = DV-IPRED
IWRES = IRES/W
```

#### **Additive plus proportional error**

```
W = SQRT(THETA(5)**2*IPRED**2+THETA(6)**2)
Y = IPRED+ERR(1)*W
IRES = DV-IPRED
IWRES = IRES/W
```

#### **-tbs**

Default not set. Invokes Transform Both Sides method, by default using the Box-Cox transformation. When -tbs is set, PsN will make the following changes to the model file before running:

1. add a THETA representing the Box-Cox LAMBDA parameter to be estimated. Default (can be changed with option -tbs\_lambda) is no lower boundary, initial estimate 1, no upper boundary.

2. A set of IF-statements will make sure log transformations are made instead of Box-Cox if the LAMBDA estimate is 0. IF statements will also handle the cases IPRED=0 and DV=0.
3. IPRED will be transformed as  $IPRED\_trans = (IPRED^{**}LAMBDA - 1) / LAMBDA$ , and the transformed IPRED used instead of IPRED.
4. The IWRES definition is changed to  $IWRES = ((DV^{**}LAMBDA - 1) / LAMBDA - IPRED\_trans) / W$
5. Any IPRED dependence in the W definition is removed.
6. The error model will be changed to an additive error model.
7. In \$SUB two fortran routines, contra.txt and ccontra\_nm7.txt are added. These files are automatically printed to the run directory. In the file ccontra\_nm7.txt that is created, the x in theta(x) points to the lambda that was added in the model file.

**-dtbs**

Default not set. Invokes Dynamic Transform Both Sides method, see [1]

The model code is modified the same way as with -tbs, see above, except that the W definition is changed using a new ZETA parameter:

1. A new parameter ZETA will be defined, to be used in the error model. The THETA for this parameter will be automatically added.
2. Any IPRED term in the W definition will be replaced with  $IPRED^{**}ZETA$ . Any THETAs that are not multiplied with IPRED in the W definition will be set to 0 FIX. If W does not depend on IPRED at all in the input model, then  $W = old\_definition$  will be replaced by  $W = (IPRED^{**}ZETA) * old\_definition$ .
3. Because lambda and zeta are correlated, estimation stability can be increased by reparameterizing and estimating

DELTA instead of ZETA, with  $ZETA=LAMBDA+DELTA$ . This reparameterization done when the user sets option `-tbs_delta`, see below.

Correspondence between `-dtbs` parameter values and common error models

| Error model                      | LAMBDA | ZETA | DELTA=ZETA-LAMBDA |
|----------------------------------|--------|------|-------------------|
| additive                         | 1      | 0    | -1                |
| proportional                     | 1      | 1    | 0                 |
| additive on log-transformed data | 0      | 0    | 0                 |

Combined error models would need to be coded manually and are not recommended for `dtbs` approach.

**`-tbs_lambda`** = *string*

Default 1 if option `-tbs` is set. Initial value string, using NM-TRAN syntax, for lambda parameter in Transform Both Sides method, e.g. “(-1, 0.5, 1)” or “0 FIX”. The string must be enclosed in quotes, double quotes on Windows and either double or single on unix, and not include any comments. If `tbs_lambda` is set then option `-tbs` will be set automatically, unless option `-dtbs` or `-tbs_zeta` or `-tbs_delta` is set. See option `-tbs` for more details.

**`-tbs_zeta`** = *string*

Default 0.001 if W does not depend on IPRED, default 1 otherwise. Initial value string, using NM-TRAN syntax, for zeta parameter in Dynamic Transform Both Sides method, for example “(-1, 0.5, 1)” or “0 FIX”. The string must be enclosed in quotes, double quotes on Windows and either double or single on unix, and not include any comments. If `-tbs_zeta` is set then option `-dtbs` will be set automatically. See option `-dtbs` for more details. Options `tbs_zeta` and `tbs_delta` cannot be used in combination.

**`-tbs_delta`** = *string*

Default not set. Initial value string, using NM-TRAN syntax, for delta parameter in Dynamic Transform Both Sides method, e.g. “(-1, 0.5, 1)” or “0 FIX”. The string must be enclosed in

quotes, double quotes on Windows and either double or single on unix, and not include any comments. If `tbs_delta` is set then option `-dtbs` will be set automatically. See option `-dtbs` for more details. Options `tbs_zeta` and `tbs_delta` cannot be used in combination.

## 4.8 Error handling

### **-abort\_on\_fail**

Default not used. If the `-abort_on_fail` option is set and one of the NONMEM runs fails, PsN will stop scheduling more runs and try to stop those that are currently running. A run is considered failed if it fails to produce a `lst` file which PsN can read. This can occur if a `nonmem` run crashes or gets killed, or if `NMtran` or compilation fails. In this context a run is not considered failed if e.g. the minimization is not successful.

### **-check\_nmtran**

Default not used. Make PsN run `NMtran` on the model file before submitting the complete `nmfe` run to a cluster/grid or forking on a local computer. This adds a bit of overhead but on a cluster this still saves time in the case of syntax errors in the model file, since the user does not have to wait for a slot on the cluster/grid before the error is detected. On a local computer the error handling is improved.

When running multiple copies of a model with different data sets, e.g. in a bootstrap, only the first model will be checked.

The `nmtran` check requires that it is the installation directory to NONMEM that is set in `psn.conf`, rather than the full path to an executable script. If the path to a script is given instead of an NM install directory the `nmtran` check will not be performed.

### **-handle\_crashes**

Default used. Disable with `-no-handle_crashes`. PsN tries to recognize if a NONMEM run has crashed for a reason where submitting the run again might help. Such a reason might be a computer crash, run being killed by system due to job

timeout, or nmfe failing to even start due to file sync errors on a cluster. If `handle_crashes` is set and the number of actual crash restarts is smaller than the setting of option `-crash_restarts`, PsN will restart those runs. The crash handling machinery is independent of the retries machinery.

If option `-handle_msfo` was also set *and* `lst-file` and `msfo-file` from the crashed run are found, PsN will automatically use the `msfo` file in `$MSFI` of a modified control stream file, and then restart the run from the state that was saved in the `msfo-file`. This functionality is useful on clusters where there is a hard run time limit for jobs.

If option `-handle_msfo` was *not* set, or if no `lst-file` or other sign that nmfe even started are found, PsN will simply try to run the same control stream again. Initial parameter estimates are not changed.

Note: If a NONMEM run is intentionally killed, e.g. with `scancel` on slurm, and PsN was started with `-handle_crashes` set, then PsN will try to restart that run. To prevent PsN restarting an intentionally killed NONMEM run, the user must either make sure `-handle_crashes` is not set or kill the main PsN process before killing the NONMEM run.

**`-crash_restarts`** = *'integer'*

Default 4. The number of times PsN will restart a crashed run. Initial estimates are not tweaked in a crash restart, see option `-handle_crashes`.

**`-handle_msfo`**

Default not used. Feature for handling resumes using `msfo` and `msfi` files.

**`-maxevals`**

Default not used. Will only work for classical estimation methods. NONMEM only allows 9999 function evaluations. PsN can expand this limit by adding an MSFO option to `$ESTIMATION`. Later when NONMEM hits the max number of function evaluations allowed by NONMEM (9999) PsN will remove initial estimates from the model-file and add `$MSFI` and

restart NONMEM. This will be repeated until the number of function evaluations specified with option `-maxevals` has been reached. Note: PsN does not change the MAXEVALS setting in the model-file, therefore the number of evaluations set on the command-line may be exceeded before PsN performs the check if the run should be restarted with `msfi` or not.

**-nice** = *'integer'*

Default 19. This option only has effect on Unix like operating systems. It sets the priority (or nice value) on a process. You can give any value that is legal for the "nice" command, likely it is between 0 and 19, where 0 is the highest priority. Execute "man nice" on the Unix system for details.

## 4.9 Advanced execute options

**-predict\_data**

Default not used. Only allowed in combination with `-predict_model`. Create copy of model with new file named `<predict_model>`, turn off estimation (MAXEVAL=0 or corresponding), change data file to `-predict_data` and then run.

**-predict\_model**

Default not used. Only allowed in combination with `-predict_data`. Create copy of model with new file named `<predict_model>`, turn off estimation (MAXEVAL=0 or corresponding), change data file to `-predict_data` and then run.

## References

- [1] Anne-Gaëlle Dosne, Martin Bergstrand, and Mats O. Karlsson. "A strategy for residual error modeling incorporating scedasticity of variance and distribution shape". In: *Journal of Pharmacokinetics and Pharmacodynamics* (2015). DOI: 10.1007/s10928-015-9460-y.