

Common options PsN

PsN 4.2.0

2014-05-21

1 Introduction

This document lists all PsN options that concern how NONMEM is run (plus a few other options). Since most PsN programs involve running NONMEM, these options are common to all those scripts. The execute program does almost nothing except running NONMEM, hence this document lists the most important options to execute.

2 Default values from source code or configuration file

PsN options can be defined in three places: in the PsN source code, in the configuration file `psn.conf` or on the command-line. The options given on the command-line always have precedence over any other values given for the same option. If an option is not given on the command-line, PsN will take values from the configuration file named `psn.conf`, see the document `psn_configuration.pdf`. Last, if the option was defined neither on the command-line nor in the `psn.conf` file, PsN will use the default value in the source code.

Please note: The default listed in this document are from the source code, but it is important to be aware that these defaults will be overridden by values given in a `psn.conf` file. For example, `crash_restarts` is set to 0 in the `psn.conf` included in the PsN release package while the PsN source code default is 4.

3 Choosing PsN version

It is possible to have several installations of PsN and use them in parallel. The default version of PsN may be changed every time a new installation is made. To check which version is the default use command

```
psn -version
```

A specific version may be called with its version number, for example

```
execute-2.3.2 -help
```

Note: When installing PsN4 on Windows all previously installed versions (installed with the same bin directory, including PsN3-versions) are fixed automatically so that they are callable with a specific version as explained above. This is a new feature of PsN4 when installing on Windows.

4 Choosing NONMEM version

If you have more than one NONMEM version installed you can use the PsN-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
```

To pick a special nonmem version use option `-nm_version`, for example

```
execute -nm_version=vi_big (more options...)
```

5 Common options

The following options will be accepted by most PsN-tools. Depending on the tool, an option may have little effect. For example, the option `-threads` will not change the execution time of `execute` unless you run more than one model. The listed default values are the default in the PsN-code. If the option is set in `psn.conf` that value will override the code default. Choices on the command-line have highest priority.

-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.

-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.

-check_nmtran

Default 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.

-clean = *'integer'*

Default 1. The clean option can take four different values: 0: means that 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: the whole NM_run directory is removed and if it is not an "execute" command, all `modelfit_dir:s` will be removed. Setting `-clean=3` will in most cases prevent resuming an interrupted run using the `-directory` option.

-crash_restarts = *'integer'*

Default 4. If a NONMEM output-file is produced but PsN is unable to read it properly it is assumed that NONMEM crashed, probably due to something in the operating system, and PsN will start the run again. But PsN will not consider it a retry and will not change initial estimates.

-cwres

Default not used. Compute the conditional weighted residuals (CWRES) for a model run. This option is disabled for NONMEM7. In NONMEM7 it is possible to request CWRES directly in \$TABLE.

-directory = *'string'*

Default `'modelfit_dirX'` for execute, and `'scriptname_dirX'` for other scripts (e.g. `bootstrap_dir1`) where X will be increased by one each time you run the script. The directory option sets the directory in which PsN will run NONMEM. You do not have to create the directory, it will be done for you. If you abort the run or if your system crashes you can use the `'-directory'` option set to the directory of the run that crashed. PsN will then not run the model-files that had finished before the crash, thereby saving time. Notice that is important that you give exactly the

same options that you gave the first time (exception npc and vpc tools, see npc/vpc manual). However, if the original run had `-clean=3` set, the information required to resume the run will in many cases have been deleted. Information on how to save information from an interrupted run in that case is found in the individual userguides.

-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. 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`.

-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, or a `defaults.pnm` file (NONMEM 7.2 or later).

-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.

-handle_crashes

Default used. Disable with `-no-handle_crashes`. PsN tries to recognize NONMEM runs that crashed for various reasons, e.g. a computer crash or a NONMEM run deliberately killed, and restart those runs without changing initial parameter estimates.

-handle_msfo

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

-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.

-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 -nonparametric_etas 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.

-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 does the check if the run should be restarted with msfi or not.

-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.

-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).

-missing_data_token = *'string'*

Default -99. missing_data_token sets the string that PsN accepts as missing data.

-niter_only = *'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 -nonparametric_etas is set. User-chosen value of NITER when estimation is turned off by setting EONLY=1. See PsN_and_NONMEM7.pdf for details.

-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.

-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`.

-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.

-nm_version = *'string'*

Default is 'default'. If you have more than one installation of NONMEM you can choose between them using the `-nm_version` option. The installations must be specified in the `psn.conf` 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.

-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. Note that before PsN4 this option was given as a comma-separated list of options that PsN would reformat. What would for PsN3 be specified as `-nmfe_options=xmloff,prdefault` must for PsN4 be specified as `-nmfe_options='-xmloff -prdefault'` on the commandline (must use double quotes on Windows) or `nmfe_options=-xmloff -prdefault` in `psn.conf`.

-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`).

-nonparametric_etas

No help available for 'nonparametric_etas'

-nonparametric_marginals

No help available for 'nonparametric_marginals'

-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 print \$OMEGA after \$THETA. To use the old print order, set option `-omega_before_pk`

-outputfile = 'string'

Default modelfilename with mod substituted for lst. The `-outputfile` option specifies the output file name for the NONMEM run. Currently this option is only valid when a single model is supplied.

-parafile = 'filename'

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

-prepend_model_file_name

Default not used. Table files by default have generic names, e.g. patab. If multiple models are run, files will be overwritten when multiple files with the same name are copied back to the same directory. This options prevents this by prepending the model file name, without extension, thus making the file names unique.

-seed = 'string'

You can set your own random seed to make PsN runs reproducible. The random seed is a string, and may include spaces if enclosed with quotes as in `-seed="123 abc"`. 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.

-shrinkage

Default not used. Calculate the shrinkage for the model run. Shrinkage is calculated as $1 - (\text{sd}(\eta(x)) / \omega(x))$ 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 modelfile 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.

-skip_data_parsing

Default set. If option is set, data files will not be parsed unless some data processing is explicitly required, for example during bootstrapping. Saves time.

-tbs

Default not set. Invokes Transform Both Sides method, by default using the Box-Cox transformation. 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/>

Model must be coded “the Uppsala way”, i.e. with the standard deviation of the residual error expressed as a THETA and SIGMA 1 FIX. 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. 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

Dosne AG, Keizer RJ, Bergstrand M, Karlsson MO. *A strategy for residual error modeling incorporating both scedasticity of variance and distribution shape*. PAGE 21 (2012) Abstr 2527.

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 `THETA`s 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.

-threads = *'integer'*

Default 1. Use the threads option to enable parallel execution of multiple NONMEM runs. 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.

-verbose

Default not used. With verbose used, PsN will print more details 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.

-version

Default not used. Print PsN version of the script being called and exit.

-warn_with_trace = *'integer'*

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.

-sde

Default not used. If you are running SDE models, you must use this option, otherwise PsN will print the records of the model file in the wrong order, and the NONMEM runs will fail.

-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.

-help

With -help execute 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.

-h or -?

Print the list of available options and exit.

-silent

Default not used. The silent option redirects all messages that are normally printed to screen to the file run_messages.txt.

Other results and log files are written to disk as usual. Nothing is printed to screen.

5.1 Retries in PsN

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` option turns on this feature in PsN, and it is on by default. Then, if the minimization is terminated 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 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 option `-prepend_model_to_lst` can be used with `execute`. Then the modelfile for the selected model with its particular initial estimates is prepended to the `lst`-file which is copied back up to the working directory. This makes it easier to see which initial estimates were actually used. The prepending is done automatically if `-nmfe` is set, so do not make the model-file be prepended twice by both setting `-nmfe` and `-prepend_model_to_lst`.

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 un-set 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

5.2 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.

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. If `-picky` is set: the run has either not finished with MINIMIZATION SUCCESSFUL, or finished with one of the picky-messages listed below.
3. If `-picky` is not set: 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`.
4. (New in version 3.4.8) If `-picky` is set: The run has finished fulfilling the picky conditions with MINIMIZATION SUCCESSFUL and none of the picky-messages, but the ofv of this run minus 'accepted_ofv_difference' is larger than the ofv of a previous run that did not fulfill the picky conditions, indicating that the current run has terminated in a local minimum.
5. (New in version 3.4.8) If `-picky` is not set: The run has finished with MINIMIZATION SUCCESSFUL, but the ofv of this run minus 'accepted_ofv_difference' is larger than the ofv of a previous run that did not finish with MINIMIZATION SUCCESSFUL, indicating that the current run has terminated in a local minimum.

After all retries have finished, PsN will select the best. If picky option was set and any run passed the picky test, the pass-picky run with the best ofv will be selected, regardless of whether any not-pass-picky run had a lower ofv. If picky was not set, or if no run passed the picky test, the selection will be based on 'corrected ofv'. For tries that did not have minimization successful the corrected ofv is equal to the ofv. For tries that had minimization successful the corrected ofv is the ofv minus 'accepted_ofv_difference'. This means that the option `accepted_ofv_difference`, default 0.5, decides how much preference should be given to runs with minimization successful. If two tries have equal corrected ofv then priority is given, in order, to a run that has minimization successful, the highest number of significant digits, or the smallest perturbation of initial estimates. If no run produces an ofv, the original model will be chosen.

-tweak_inits

Default used, 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 and smaller than 1. 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. 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 NON-MEM 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'*

Important note: The meaning of this option has changed between PsN-3.4.7 and PsN-3.4.8. Default value 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.

5.3 Options for grid/cluster execution

-run_on_torque

Default not used.

-torque_queue

Default empty. Maps to `-q` in `qsub`.

-torque_prepend_flags

Default empty. Only valid with `-run_on_torque`. The `-` signs must be included in the string. The flags/options will be prepended to standard `'-N jobname -q torque_queue'` where `jobname` is `'psn:'`<modelfile> and `torque_queue` is read from `psn.conf` or set on the commandline. If multiple options are given using `-torque_prepend_flags` on the commandline, they must be separated by spaces (as in `qsub`) and the whole list enclosed by quotes. If `torque_prepend_flags` is set in `psn.conf`, there should be no quotes.

- run_on_lsf**
Default not used. PsN connects with Platform Load Sharing Facility (LsF). PsN submits nmfe directly with bsub.
- lsf_job_name = *'string'***
Maps to bsub option -J. Sets the name of the LSF job name of every NONMEM run, they all get the same name (e.g. all samples of a bootstrap get the same name, all candidate models in an scm get the same name). If not set, and option -run_on_lsf is set, the default job name is the model file name, meaning that each bootstrap sample gets its own name, each scm candidate model gets its own name, etc.
- lsf_project_name = *'string'***
Maps to bsub option -P. Use lsf_project_name to assign a project name to your LSF runs.
- lsf_queue = *'string'***
Maps to bsub option -q. lsf_queue specifies which LSF queue PsN should submit NONMEM runs to and is used in conjunction with -run_on_lsf
- lsf_resources = *'string'***
Maps to bsub option -R. lsf_resources specifies which LSF resources is required when submitting NONMEM runs.
- lsf_ttl = *'string'***
Maps to bsub option -c. lsf_ttl sets the maximum time a NONMEM run should be allowed to run on the LSF grid.
- lsf_options = *'string'***
LSF jobs are submitted using bsub and all PsN's LSF related options are translated to corresponding bsub options. If a specific bsub option is not available through any of the other lsf-specific options, -lsf_options can be used to pass any set of options to bsub. The string must be enclosed in quotes if it contains spaces.
- lsf_sleep = *N***
Default 3. Wait for this many seconds after bsub submission, before continuing running PsN.

- run_on_sge**
Default not used. Submit NONMEM runs to Sun Grid Engine.
- sge_queue = *'string'***
Default empty. Only valid with -run_on_sge. Maps to qsub option -q
- sge_resource = *'string'***
Default empty. Only valid with -run_on_sge. Maps to qsub option -l
- sge_prepend_flags = *'string'***
Default empty. Only valid with -run_on_sge. The - signs must be included in the string. The flags will be prepended to standard flag set in qsub call. If multiple flags are given using the option on the commandline, they must be separated by spaces (as in qsub) and the whole list enclosed by quotes. If the option is set in psn.conf, there should be no quotes.
- run_on_ud**
Default not used. PsN connects with United Devices Grid MP. With -run_on_ud PsN will submit to the UD grid with parameters defined in the "uduserconf" file.
- run_on_zink**
Default not used. Experimental.
- run_on_slurm**
Default not used. Use slurm queueing system.
- slurm_account = *string***
Default empty. Only valid with -run_on_slurm, then optional. Maps to sbatch option -A. This option was called -slurm_project in PsN3, but it has been renamed to match the slurm system documentation where the term is 'account'.
- slurm_partition = *string***
Default empty. Only valid with -run_on_slurm, then optional. Maps to sbatch option -p.

-max_runtime = *'string'*

Default not used. Only allowed with `-run_on_slurm`. A limit on how long a slurm run may go on before being aborted (option `-t` to `sbatch`). Format is either minutes, e.g. `-max_runtime=10`, or hours:minutes:seconds, e.g. `-max_runtime=4:0:0`, or days-hours, e.g. `-max_runtime=3-0`

-send_email

Default not set. Only used with `-run_on_slurm` and `-email_address` in combination, otherwise ignored. Used for `sbatch` options `-mail_user=<email_address>` `-mail_type=ALL` or `END`.

-email_address = *string*

Default not set. Only used with `-run_on_slurm` and `-send_email` in combination, otherwise ignored. Used for `sbatch` options `-mail_user=<email_address>` `-mail_type=ALL` or `END`.

-slurm_prepend_flags = *'string'*

Default empty. Only valid with `-run_on_slurm`. The `-` signs must be included in the string. The flags will be prepended to standard flag set in `sbatch` call. If multiple flags are given using the option on the commandline, they must be separated by spaces (as in `sbatch`) and the whole list enclosed by quotes. If the option is set in `psn.conf`, there must be no quotes.