

Common options, defaults and program versions in PsN

2013-10-14 PsN 3.6.3

Introduction

The various PsN scripts share many building blocks, and many command-line options are common to all PsN scripts. The command `psn -h` list all available scripts, and the `psn_options -h` lists all common PsN options. By using `-help` instead of `-h` more details are printed.

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.

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.

Which version is the default? Use command
`psn -version`

A specific version may be called with its version number, for example (on Unix type systems)
`execute-2.3.2 -help`

When calling a *non-default* version on windows it is currently necessary to use
`perl <full_path_to_PsN_executable><name_of_PsN_script> <PsN options>`
for example

`perl C:\Perl\bin\execute-2.3.2 -help`

Choosing NONMEM version

You can use the PsN-option `-nm_version` to use different versions of NONMEM. To use a NONMEM version it must be 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`

You can use a special nonmem version listed in `psn.conf` via the option `-nm_version`, for example

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

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.

Optional input

-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, if NMtran or compilation fails, or if incorrect perl settings in psn.conf prevents NMtran from being initiated at all. 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.
-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.
-compress	Default not used. PsN will compress the contents of 'NM_runX' to the file 'nonmem_files.tgz' if the -compress option is used and if you have the archive and compress programs tar and gzip installed. If you use the -clean options, run files will be removed before the compression.
-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).
-drop_dropped	Default not used. If there are drop columns in your control file and -drop_dropped is set, PsN will remove those columns from the data set used internally. It saves both disk-space and conserves memory usage. Note that PsN does NOT alter your original data set, only those used internally in PsN.
-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_data_files ='extra_data1.dta, COLUMN1, COLUMN2'	Default not used. NONMEM only allows 20 column datasets, but PsN can add code to control files that reads extra data columns from a separate file. To use this feature you must divide the original file into two using the script single_valued_columns (distributed together with PsN). Provided that each of the two new files contain at most 20 columns, you write the name of the file with multiple valued columns as the data file in the NONMEM modelfile. On the command line you then specify a comma separated list with -extra_data_files. The first element in the list is the name of the file with single valued columns, and the rest of the list is the column headers of the extra data file. If any of the two new files generated by single_valued_columns contains more than 20 columns, this feature cannot be used.
-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-separ ated 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. NM7.2 can, in addition to the lst-file, generate 11 files. PsN will always copy the lst-file back to the calling directory. The option -nm_output decides which of the 11 additional files should also be copied back. The default is none, but the template psn.conf distributed with PsN has ext,cov,coi,cor,phi set. Note that NM output files which are not copied to the calling directory can still be found inside the run directory.
-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 (in sourcecode) not used but set as default in template psn.conf. New feature in PsN3. Invoke nmfe6 or nmfe7 (not nmfe5) from within PsN instead of doing stepwise compiling and execution. PsN will look for nmfeX, where X is the NONMEM version number specified in psn.conf, in first the /util then the /run and last the /. subdirectory of the NONMEM installation directory specified in psn.conf, and call the first instance found. The compiler settings in psn.conf are not needed. On some computer architectures it may still be necessary to perform stepwise compiling and execution. Option nmfe is set automatically if the user sets for example run_on_sge_nmfe, run_on_lsf_nmfe or run_on_slurm.
-nmfe_options=<list>	Only relevant if option -nmfe or -nmqual is set (note that -nmfe is sometimes set automatically, see option -nmfe) and NONMEM7.2 or later is used. A comma-separated list of options that will be passed on to the nmfeX script (NMQual8 also uses this script). The options must be given without the - signs. PsN will add these, for example -nmfe_options=prsame,xmloff will append -prsame -xmloff to the nmfe call. PsN will not check that the options are appropriate given existing compiled files and properties of the model and data, nor check that the option is one that will be recognized by nmfe. It is possible to set -nmfe_options=licfile=somenewlicfile.lic but then the user must also set -extra_files=somenewlicfile.lic. Same for nmexec. If an absolute path to the license file is given then option -extra_files is not needed.

-nmqual	Default not used. New feature in PsN-3.1.2. Invoke an NMQual generated perl script <nmqualscript.pl> from within PsN instead of doing stepwise compiling and execution. PsN will look for <nmqualscript.pl>, where <nmqualscript.pl> must include a full path and must be specified in the [nm_versions] section of psn.conf. Note that the scriptname itself must also be included in the path in psn.conf, not only a directory as when -nmfe is set. The compiler settings in psn.conf are not needed. If NMQual8 PsN will assume that a lst-file will be produced, and handle that the normal way. For earlier NMQual versions PsN will assume that a file named OUTPUT or output, containing the standard text format NONMEM output, can be found in the run directory after <nmqualscript.pl> has been called. To mimic the behaviour of nmfe6 and nmfe7, PsN will create a lst-file by copying OUTPUT, prepending the model file and (nmfe7 only) FMSG, and then use the lst-file in all subsequent steps as if it had been generated by nmfe. The xml-format output-file will not be read by PsN, and deleted if clean>1. Option prepend_model_to_lst should not be used with the nmqual option, then the model would be prepended twice to the lst-file. Option -nmqual cannot be set together with option -nmfe. Note that -nmfe is sometimes set automatically, see option -nmfe.
-nmqual_xml=string	Default 'log.xml'. Only relevant with option -nmqual and with NMQual8 and later. This option can be used to set the path and name to the xml-file to which NMQual8 should write results. The file set using this option will be used when PsN runs perl path/autolog.pl nmqual_xml_including_path run ce (...)
-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.
-parafilename='filename'	NONMEM 7.2 (or later version) parafilename. Can only be used if option -nmfe or -nmqual is set. Note that -nmfe is sometimes set automatically, see help for -nmfe. Appends "-parafilename=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.
-rerun='integer'	Default 1. PsN can redo or resume a run using information in PsN run directory(see documentation for -directory). It is called a rerun. During a rerun PsN will consider to redo parts of the run. With the -rerun option you can control which parts will be redone. With rerun set to 1 PsN will rerun any model with a missing list file. Notice that every "retry" (see the

	documentation for -retries and -min_retries) will be considered for a rerun. This means you can change the value of the -retries and -min_retries options if you like more or less retries. Setting -rerun to 0 means that PsN will not check for missing or incomplete "retry" list files. This is useful if you have one or more run model-files and you wish to have a PsN raw_results file. You do a run with them as arguments and specify -rerun=0, PsN will not do any NONMEM run, but produce useful output summary. You can also set -rerun to 2, and PsN will ignore any existing list files and rerun everything, creating raw_results from the new listfiles. Default Perl generated random number. You can set your own random seed to make PsN runs reproducible. The random seed will be used with the -retries='integer' option and with e.g. the bootstrap, vpc, npc and nonpb scripts.
-seed='string'	
-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. Warning: PsN-3.2.4 and earlier versions assumes that all parameters are estimated for all subjects. Therefore incorrectly high shrinkage values can be presented in the raw_results file for parameters not estimable for certain individuals, e.g. drug effect parameters in the presence of subjects receiving only placebo treatment. In NONMEM7 eta shrinkage values are output in the lst-file, and PsN will present these instead. PsN-3.2.5 and later assumes that ETAs which are exactly 0 have not been estimated, and excludes these from the computations. PsN-3.2.5 and later never uses 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.
-skip_data_parsing	Default set. Automatically unset if -drop_dropped, -extra_data_files or -wrap_data is used, and with bootstrap and cdd scripts. If option is set, data files will not be parsed, only dummy data objects will be created. Saves much 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.thtinfo.com/

Model must be coded “the Uppsala way”, i.e. with IWRES and W 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_param) 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
$$\text{IPRED_trans} = (\text{IPRED} ** \text{LAMBDA} - 1) / \text{LAMBDA},$$
and the tranformed IPRED used instead of IPRED.
- 4) The IWRES definition is changed to
$$\text{IWRES} = ((\text{DV} ** \text{LAMBDA} - 1) / \text{LAMBDA} - \text{IPRED_trans}) / W$$
- 5) The error model is changed to additive, if not already additive:
$$Y = \text{IPRED_trans} + \text{EPS}(1) * W$$
- 6) In \$SUB two fortran routines, contra.txt and ccontra_mod2.txt are added. These files are automatically printed to the run directory. In the file ccontra_mod2.txt that is created, the x in theta(x) points to the new theta that was added in the model file.

-tbs_param=<string>	Default not set. Initial value string, using NM-TRAN syntax, for parameter in Transform Both Sides method, e.g. '(-1, 0.5, 1)' or 'O FIX'. The string must be enclosed in single quotes and not include any comments. If tbs_param is set then option -tbs will be set automatically. See option -tbs for more details.
-threads='integer'	Default 1. Use the threads option to enable parallel execution of multiple NONMEM runs. On a desktop computer it is recommended to set -threads to 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. The -threads option will be ignored if you run on a grid system, since grids have their own scheduling algoritms.
-unwrap_table_files	Default not used. Ensures that all table rows are on a single line in the table file.
-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 successfull 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.
-wrap_data	Option wrap data has been disabled. Use \$SIZES in NONMEM to handle more than 50 items in \$INPUT.

<code>-warn_with_trace='integer'</code>	Default not used. If <code>-warn_with_trace</code> is set, PsN will print a stack trace for all error and warning messages. This is only for developers.
<code>-sde</code>	Default not used. If you are running SDE models, you must use this option, otherwise PsN will print the records of the modelfile in the wrong order, and the NONMEM runs will fail.
<code>-debug='integer'</code>	Default 0. The <code>-debug</code> 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.
<code>-help <option_name></code>	With <code>-help</code> 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.
<code>-h -?</code>	Print the list of available options and exit.
<code>-silent</code>	Default not used. The silent option redirects all messages that are normally printed to screen to the file <code>run_messages.txt</code> . Other results and log files are written to disk as usual. Nothing is printed to screen.

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` or `-run_on_sge_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 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

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 $init_i = init_0 + rand_uniform(+/-0.1*init_0)$ where $init_0$ are the initial estimates of the original run. The updating procedure makes sure that boundary conditions on the parameters are still valid. For this option to have effect, the -retries option must be set to a number larger than zero.
-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: 0ESTIMATE OF THETA IS NEAR THE BOUNDARY 0PARAMETER ESTIMATE IS NEAR ITS BOUNDARY 0R MATRIX ALGORITHMICALLY SINGULAR 0S 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'

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.

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:<model>' 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_mosix	Default not used. Only implemented and tested for Unix type systems. PsN will start the perl processes running NONMEM with 'mosenv -e perl...' instead of the default 'perl...'. (The name of the perl binary is configurable in psn.conf.)
-run_on_lsf	Default not used. It is recommended not to use this option but to use -run_on_lsf_nmfe instead. PsN connects with Platform Load Sharing Facility (LsF).
-run_on_lsf_nmfe	Default not used. Recommended instead of run_on_lsf, faster and more stable. PsN connects with Platform Load Sharing Facility (LsF). PsN submits nmfe directly with bsub (instead of as with -run_on_lsf submitting a perl process nonmem.pm which in turn invokes nmfe). When run_on_lsf_nmfe is set option -nmfe will be set automatically.
-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_nmfe 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 <code>-run_on_lsf</code>
<code>-lsf_resources='string'</code>	Maps to bsub option <code>-R</code> . <code>lsf_resources</code> specifies which LSF resources is required when submitting NONMEM runs.
<code>-lsf_ttl='string'</code>	Maps to bsub option <code>-c</code> . <code>lsf_ttl</code> sets the maximum time a NONMEM run should be allowed to run on the LSF grid.
<code>-lsf_options='string'</code>	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, <code>-lsf_options</code> can be used to pass any set of options to bsub. The string must be enclosed in quotes if it contains spaces.
<code>-lsf_sleep=N</code>	Default 3. Wait for this many seconds after bsub submission, before continuing running PsN.
<code>-run_on_sge</code>	Default not used. Use Sun Grid Engine queueing system.
<code>-run_on_sge_nmfe</code>	Default not used. A slimmed down alternative to setting <code>-run_on_sge</code> and <code>-nmfe</code> in combination. With this option set, the qsub command will be on nmfe directly, instead of on 'perl nonmem.pm' which in turns calls nmfe. This removes much overhead (no perl process submitted with qsub) at the cost of some of PsN:s error handling implemented in the nonmem.pm module. Setting this option bypasses nonmem.pm completely. The options <code>-sge_queue</code> , <code>-sge_resource</code> and <code>-sge_prepend_flags</code> work the same way as with <code>run_on_sge</code> . When <code>run_on_sge_nmfe</code> is set option <code>-nmfe</code> will be set automatically.
<code>-sge_queue='string'</code>	Default empty. Only valid with <code>-run_on_sge</code> . Maps to qsub option <code>-q</code>
<code>-sge_resource='string'</code>	Default empty. Only valid with <code>-run_on_sge</code> . Maps to qsub option <code>-l</code>
<code>-sge_prepend_flags='string'</code>	Default empty. Only valid with <code>-run_on_sge</code> . The <code>-</code> 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.
<code>-run_on_ud</code>	Default not used. PsN connects with United Devices Grid MP. With <code>-run_on_ud</code> PsN will submit to the UD grid with parameters defined in the "uduserconf" file.
<code>-run_on_zink</code>	Default not used. Experimental clustering on Windows machine. Non-functional 2008-08-01.
<code>-run_on_slurm</code>	Default not used. Use slurm queueing system. When <code>run_on_slurm</code> is set option <code>-nmfe</code> will be set automatically.
<code>-slurm_project=<string></code>	Default empty. Only valid with <code>-run_on_slurm</code> , then optional. Maps to sbatch option <code>-A</code> .
<code>-max_runtime='string'</code>	Default not used. Only allowed with <code>-run_on_slurm</code> . A limit on how long a slurm run may go on before being aborted (option <code>-t</code> 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 should be no quotes.