PsN configuration

PsN 4.6.0

Revised 2016-05-10

1 The quick and easy way

Let the installation script create psn.conf for you. If you have already installed PsN and did not let the script create psn.conf but have changed your mind now, redo the PsN installation and have psn.conf automatically generated this time. If you will run PsN locally on your computer with a standard NONMEM installation you are then done.

An alternative to letting the installation script create a new psn.conf is to copy psn.conf from a previous working PsN installation to the new installation directory. If the old psn.conf was from PsN3 you need to edit default settings of nmfe_options, if set in the old psn.conf, because the syntax of this option has changed between PsN3 and PsN4.

If you have a NMQual8 installation of NONMEM and want to always run NONMEM via the autolog script then you need to replace nmfe=1 with nmqual=1 in the `[default_options]` section in psn.conf. You should also consider adding log.xml to the nm_output list of file extensions.

If you have other needs then you need to read the rest of this document.

2 Overview

The configuration file psn.conf is required to make PsN run correctly. In the configuration file the NONMEM installation directory is specified, together with essential version information. In psn.conf it is also possible to specify personal default values, see more information below. Comment lines in

psn.conf start with a semicolon (;). Throughout the default psn.conf distributed with PsN there are examples of settings for Unix and Windows. The user must review the settings, add and remove semicolons, and change selected paths.

3 Configuration file location

PsN will look for psn.conf at two different places, in PsN installation directory and in either user's home-directory (UNIX) or the user's desktop (Microsoft Windows). If a user-level configuration file is found, the settings in this file will override the settings in the system-wide (PsN installation directory) configuration file. This is important to remember when trying to sort out a PsN-related problem.

4 Sections in psn.conf

The organizing of settings in psn.conf is important. They must be set in the correct section. A section starts with `[section_name]` (a new line where square brackets enclose the section name), for example `[nm_versions]`. The exception is the first section which starts at the beginning of the file and has no name. A section ends with `[next_section_name]` or the end of the file.

5 Setting defaults for command line options in psn.conf

The user can override source code defaults by setting options in the appropriate section in psn.conf. Settings on the command line will override settings in psn.conf.

The option names used in psn.conf are the same as would be used on the command line, and documentation can be found in the document common_options.pdf. Some options do not take any value on the commandline, for example -run_on_sge, but if set in psn.conf those options must also have a value (1 or 0). If such an option is set in psn.conf, for example -run_on_sge = 1, the option can be disabled on the commandline by using -no, for example -no-run_on_sge. The -no prefix cannot be used in psn.conf, instead set

the option to 0.

The syntax in psn.conf is as follows: Options values to be used for all tools must be set in the section `[default_options]` On each row in the section the option name (without leading minus-sign) comes first. Then there an equal sign and then the value. Spaces may be added around the equal sign. Options which on the commandline can be set just by giving the option name, e.g. abort_on_fail, must in psn.conf be given the value 1. Examples:

```
abort_on_fail = 1
nmfe = 1
```

Options with values:

```
threads = 5
sge_prepend_flags = -V
```

Options for specific scripts, e.g. llp, can be set in the sections `[default_<scriptname>_options]` , for example `[default_llp_options]`.
New sections may be added, example:
`[default_sse_options]`
Defaults set in a script-specific section will override settings in `[default_options]`.

5.1 -nm_version dependent defaults

It is possible to set different defaults for different NONMEM versions. This functionality is intended for advanced users only. Each NONMEM version specified in psn.conf has a name (see section 'Essential NONMEM information') that can be used on the commandline to invoke this version, e.g. -nm_version=nm72. When (for example) -nm_version=nm72 is set on the (for example) vpc commandline, PsN will look for sections `[default_options_nm72]` and `[default_vpc_options_nm72]` and read options from there in addition to from the regular `[default_options]` and `[default_vpc_options]`. The option -nm_version *must be set on the commandline* to invoke this feature. NONMEM-version specific defaults will *not* be used if option nm_version is only set in psn.conf.

PsN will look for option settings in the following order. Settings on the commandline have precedence over settings in `[default_scriptname_options_nmversion]` (if present), which have precedence over `[default_scriptname_options]` (if present), which have precedence over `[default_options_nmversion]` (if present), which have precedence over `[default_options]` (if present).

6 Warning

If running PsN on unix but editing psn.conf on Windows, make sure to save psn.conf in unix format or to convert after editing.

7 Essential NONMEM information

It is necessary to specify NONMEM installation information in psn.conf. This is done in the section

```
[nm_versions]
```

in psn.conf. The format is

```
name=installation_directory,version_number
```

The NONMEM version number should be major.minor, as in 7.2 for example. The major version number and the subversion number should be entered with a dot in between, no spaces.

Some examples:

```
[nm_versions]
;with NONMEM7.2
7_1=/opt/NONMEM/nm71,7.1
7_2=/opt/NONMEM/nm72,7.2
;Windows example
;7_2=c:\nm7_2,7.2
```

By default, PsN will use the NONMEM installation identified by the name 'default'. To use other installations with PsN, use the "-nm_version" command line option, for example -nm_version=vi_big

It is optional to add a comma and a description of the NONMEM installation after the version number. Example:

```
[nm_versions]
;with NONMEM7.2
7_1=/opt/NONMEM/nm71,7.1,NM 7.1 using gfortran 4.4.7
7_2=/opt/NONMEM/nm72,7.2,NM 7.2 using gfortran 5.1.1
```

To check which versions are defined in psn.conf without opening the file, use the command psn -nm_versions This command will also show the optional description text after the version number.

If the user wants to invoke nmfe using a wrapper, or use a modifed nmfe script with a different name, the name of the alternative script including the full path can be set under [nm_versions] instead of the installation directory. PsN will check that the file exists and is executable. Using a wrapper is useful e.g. when environment variable changes are needed prior to nmfe execution.

For NMQual8 installations where the user want to invoke NONMEM using autolog.pl (instead of using nmfeX) the NONMEM installation directory should be set to exactly the same path as the 'target' alias set in the xml file used for installation. For example, if the xml file used at installation defines

```
<alias id='target'>C:\nm73</alias>
```

then psn.conf should read

```
[nm_versions]
73=C:\nm73,7.3
```

The PsN installation script can handle setting [nm_versions] automatically.

Note that -nmqual must be set on the command-line or in psn.conf for PsN to run NONMEM using the autolog script.

7.1 NONMEM information for portable PsN

This functionality is intended for advanced users only. The `[nm_versions]` section in psn.conf can list the name of the nmfe executables instead of the full path to the NONMEM installation directories *if at runtime* the nmfe executable, chosen by default or with option -nm_version, is in the user $PATH. Example:

```
[nm_versions]
default=nmfe73,7.3
nm72=nmfe72,7.2
nm712=nmfe7,7.1
```

PsN will exit with an error message if the executable chosen at runtime is not in the user $PATH.

The names of the executables can be given without the extension on Windows *if* the extension is either '.bat' or '.exe'.

The above form of `[nm_versions]` must be manually created in psn.conf. It cannot be automatically generated by the setup script.

8 Running nmfe from within PsN

PsN will only run NONMEM via the nmfe script, an alternative script set by the user, or an NMQual autolog.pl script. If option nmfe is set (the default), PsN will look for nmfeX, where X is the NONMEM version number specified in the nm_versions section (X is 72 if NONMEM7.2 is used), in first the /run then the /util and last the /. subdirectory of the installation directory specified in psn.conf, and call the first instance found. Note that it is possible to use nmfeX scripts located in any directory specified in the [nm_versions] section, not just a standard NONMEM installation directory, and that a wrapper can also be used, see section Essential NONMEM information.

9 NMQual and PsN

PsN4 supports NMQual8 (no older NMQual versions). The user must set option -nmqual, either in psn.conf or on the command-line, and the NONMEM installation directory must be properly defined in psn.conf (see section Essential NONMEM information above). PsN will look for autolog.pl in the nmqual subdirectory of the NONMEM installation directory, and return an error if not found. PsN will use the log.xml file in the nmqual subdirectory of the NONMEM installation directory as input to autolog.pl, so it is important that nmqual/log.xml exists and that the containing NONMEM installation information has not been changed after the original NONMEM installation via NMQual.
The full command used by PsN is
`perl /path/autolog.pl /path/log.xml run ce /full/path/workdir psn` (extra NM options) or, if option -parafile is set, PsN will copy the parafile to psn.pnm in the NM_run directory and use the command
`perl /path/autolog.pl /path/log.xml para ce /full/path/workdir psn` (extra NM options)

The user is not required to modify log.xml in any way when running with

PsN4. When running PsN3 with NMQual the user needed to modify the 'run' section of log.xml, but with PsN4 log.xml should be left as is.

Important note: The extra NM options above are options set with e.g. -nmfe_options or -nodes, but unless the do-on-run block of the log.xml file uses 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.

To facilitate debugging, the command used to invoke autolog.pl will be printed to the file nmqualcommand in the NM_runX subdirectory when PsN is run locally on *nix or Windows. When -run_on_sge or -run_on_slurm is used, the command will be printed to qsubcommand/sbatchcommand (also when -nmfe is used).

If the user does not set option -nmqual, but sets -nm_version to an NMQual installation of NONMEM, then PsN will look for an nmfe script in the installation directory as described in section Running nmfe from within PsN.

10 Compiler configuration

In PsN4 no compiler instructions can be set.

11 Job submission delays

When more than one model will be run in parallel (threads > 1 and more than one model to run), it is possible to make PsN pause between submissions of new models. The relevant configuration variables are `min_fork_delay` and `max_fork_delay`, which if used must be set *before* the first section in psn.conf, i.e. before the first `[section_name]` header. `max_fork_delay` is the maximum delay in seconds, and `min_fork_delay` is the step-length in micro-seconds between each check if output from the previous model in the list has appeared. If output from previous run is visible the pause ends. Example:

```
min_fork_delay=100000
max_fork_delay=1
```

which will give at most 1 second pause in steps of 0.1 seconds. Of course, if `min_fork_delay` is set larger than $10^6 \cdot$`max_fork_delay` then the delay will

be longer than `max_fork_delay` seconds. This is a method to make the delay independent of the output generation from other models.

12 Polling interval

By default, PsN will check for finished jobs every 1 second. By setting `job_polling_interval` in psn.conf before the first `[section_name]` header, the interval can be changed. However, it cannot be set to 0. Example:

```
job_polling_interval=2
```

which will make PsN check for finished jobs every 2 seconds.