PsN configuration

2013-05-28 PsN 3.6.2

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 install you are then done.

If you have a previous PsN 3 installation then you can copy psn.conf from the old installation to the new one.

If you have a NMQual installation of NONMEM you need to read the section "Running NMQual-generated perl scripts from within PsN", change one or a few lines in the [nm_versions] section of psn.conf, and possibly redo the NMQual installation of NONMEM.

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

Introduction

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.

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.

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.

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 common_options_defaults_versions_psn.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, just set the heading default_<scriptname>_options in square brackets. Example:

```
[default_sse_options]
```

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.

Changes in PsN 3.4.2 to handle NONMEM 7.2

The new features of NONMEM 7.2 makes it necessary to set also the subversion number in psn.conf, to enable PsN to handle NM7.1 and NM7.2 differently. For older PsN versions the NONMEM version number set in [nm_versions] must only be 5, 6 or 7, but starting with 3.4.2 it may include subversion number, e.g. as in 7.1 or 7.2. Subversion number is not required for NM7.1 or older, but correctly specifying subversion number is necessary to be able to use NM7.2-specific features. The major version number and the subversion number should be entered with a dot in between, no spaces.

So far NONMEM7.2 is supported if option -nmfe, -run_on_lsf_nmfe, -run_on_slurm, -run_on_sge_nmfe or -nmqual is set.

```
[nm_versions]
;subversion number can and should be included with PsN-3.4.2 and later in combination
;with NONMEM7.2
7_1=/opt/NONMEM/nm71,7.1
```

7_2=/opt/NONMEM/nm72,7.2

Perl executable

For users with Windows7 and 64bit, and for Windows users in general, this is an important section to read *unless* you let the installation script create psn.conf.

In psn.conf, in the top section that does not have a header, it is sometimes necessary to set

1) remote_perl=some_path_to_perl
where the default is simply 'perl'. It is not used with run_lsf_nmfe, run_on_sge_nmfe or run_on_slurm, but it is used together with run_on_lsf, run_on_torque, run_on_zink and run_on_sge. It is the path to the Perl version for running nonmem.pm (where the nonmem runs are started but no setup or analysis is done) on the remote system.

2) perl=some_path_to_perl
where the default is C:\Perl\bin\perl.exe on Windows and perl on Unix. This is the Perl version used to run nonmem.pm when using local execution. If you are using Windows, the Perl version should be the same as the one chosen during the installation using the setup script (the choice for "Path to perl binary used to run Utilities" in the setup).
For users with Windows7 and 64bit, the default setting in psn.conf is likely to not work, most likely the correct setting is C:\Perl64\bin\perl.exe. The installation script will probably find the correct version, pay attention to what it suggests for "Path to perl binary used to run Utilities" and copy to psn.conf.

All other parts of PsN (setting up runs, bootstrapping datasets, extracting results, performing analyses...) is done using the perl version set during installation. This version is hard
coded into the programs. Obviously this perl version can be different for different versions of PsN.

If there are spaces in the Windows file paths, e.g. C:\Program files\ACTIVEPERL.588.EN\bin\perl.exe then the following trick must be used: Take the first 6 characters of the path level containing a space (e.g. Progra), append a tilde, then a number representing the instance of the replaced pattern, i.e. 1, giving C:\Progra~1\ACTIVEPERL.588.EN\bin\perl.exe. If you have more spaces in the same path (e.g. C:\Program Files\ACTIVE PERL\version xxx\) the replacement would be
C:\Progra~1\ACTIVE~1\versio~1\

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 must only be 5, 6 or 7 for older PsN versions, but must include subversion number, e.g. as in 7.2, starting with PsN-3.4.2. The major version number and the subversion number should be entered with a dot in between, no spaces.

Some examples for PsN 3.4.2 and later:

```
[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 To check which versions are defined in psn.conf without opening the file, use the command

```
psn -nm_versions
```

If the user wants to invoke nmfe using a wrapper, the name of the wrapper 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 NMQual installations where the user want to invoke NONMEM using an NMQual generated Perl script (instead of using nmfeX) the information needed under [nm_versions] is slightly different, see section Running NMQual-generated Perl scripts from within PsN.

Running nmfe from within PsN

PsN will only run NONMEM via nmfe script or an NMQual script. Option nmfe is set as default in the template psn.conf (but not in the source code). 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.

Running NMQual-generated perl scripts from within PsN

A feature of PsN-3.1.2 and later is to invoke an NMQual generated perl script <nmqualscript.pl> from within PsN instead of doing stepwise compiling and execution or running nmfeX. To try the feature use the option -nmqual (no value). 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 script name itself must also be included in the path in psn.conf, not only a directory as when -nmfe is set. It is possible to both specify standard NONMEM installation directories and NMQual scripts in psn.conf. The value of the command line option nm_version will decide which information is used. Example:

```
[nm_versions]
; Unix examples
default=/opt/NONMEM/test/nmvi.pl,6
6nmqual=/opt/NONMEM/test/nmvi.pl,6
72nmqual=/opt/NONMEM/nm72/test/autolog.pl,7.2
```

```
execute run1.mod -nmqual -nm_version=6nmqual (NONMEM is run via nmvi.pl)
```

To use option -nmqual it is necessary to set system paths and/or aliases so that no path to the perl executable is needed, it must be possible to use only 'perl'. It is currently not possible to set a path to the Perl executable for NMQual.

NMQual8 and NM7.2

The user must set any NM7.2-specific options via -parafile and -nodes and/or -nmfe_options (NMQual 8 invokes a modified version of nmfe72). With NMQual8 (necessary for NM7.2) the option -nmqual_xml is needed, the default is log.xml. Also, the installation of NONMEM must have been performed using a slightly modified nm72.xml. PsN users should

1. change aliases in nm72.xml just as for any NMQual8 installation, see NMQual8 documentation.
2. Also change this

```
<do on='run' in='$_2'>target/util/nmfe72 $_1 $_3.ctl $_3.lst</do>
```

to this:

```
<do on='run' in='$_2'>target/util/nmfe72 $_1 $_3.mod $_3.lst</do>
```

in nm72.xml (i.e. change the expected suffix of control streams from .ctl to .mod) and then perform a standard install, per NMQual8 documentation.

PsN itself will then invoke the installation like this:

```
perl full_path_to_autolog.pl full_path_to_log.xml run ce path_to_workdir psn (extra nmfe72 options)
```

where the full path to autolog.pl, including the filename itself, must be set in psn.conf (see above) and full_path_to_log.xml is set via the command-line, default log.xml in the directory from which PsN is called.

Compiler configuration

From PsN 3.6 and on it is not possible to set compiler instructions in PsN.