

Release Notes for NONMEM 7.1.0

1 October 2009

NONMEM 7 Release Notes will be updated periodically and the latest version can be found at ftp://nonmem.iconplc.com/Public/nonmem7/Release_Notes_Plus/

I. The NONMEM License File

NONMEM 7.1.0 uses a license file, nonmem.lic, to alert the user regarding the license expiration date. Program execution cannot be initiated after the stated expiration date unless replaced by a new license file with a new expiration date obtained from Icon Development Solutions.

The NONMEM 7.1.0 CD is shipped with a temporary license file with an expiration date of 15 OCT 2009. When the CD is received, the contact person should send an e-mail message to IDSSOFTWARE@iconplc.com requesting a permanent license. The message should include the company name and department or, if an individual license, the individuals name. The new license file will be provided as an e-mail attachment. This file should replace the previous license file in the directory named license.

II. Known Problems

The following 8 problems were known at the time the NONMEM 7.1.0 CD's were reproduced. These items are also included in the "readme_7.1.0" file on the CD.

1) Installation on an HP 64-bit machine with Windows XP Professional x64 and Intel Fortran 9.1 64-bit was successful without optimizations but failed with the default optimizations due to /Ob1gyti. The workaround is to copy the contents of the CD to a directory. In SETUP7.bat change the line

```
if "%f%" == "ifort" set op=/Gs /nologo /nbs /w /Ob1gyti  
/Qprec_div /4Yportlib /traceback
```

to

```
if "%f%" == "ifort" set op=/Gs /nologo /nbs /w /Qprec_div  
/4Yportlib /traceback
```

save SETUP7.bat and repeat the installation from the directory.

2) Installations on Windows Vista, 32- and 64-bit, using Intel Fortran have been successful. However, installations on Windows Vista, 64-bit, using gfortran or g95 and on Solaris using the Sun Studio 12 version of f95 have failed. (But see update below.) Installations on Windows Vista, 32-bit, using gfortran and g95 have not been tested.

3) Run-time control keys do not work for Mac OS X with g95 and gfortran.

4) The following message may occur in Linux, Mac OS X or Solaris, and indicates that there is insufficient memory to run NONMEM (typically, with 'big ' sizes):

```
nmfe7: line 65: 12232 Killed  ./nonmem $1 $3 -licfile=$lfile
```

Edit resource/SIZES.f90 and supply smaller values . Retype the SETUP7 command, with options
“same rec” instead of “big norec”

5) If the dimension of SIGMA in the model is greater than 6 then the MAXPTHETA and LTH variables in SIZES must be modified.

$MAXPTHETA \geq NTHETA + NSIGMA * (NSIGMA + 1) / 2$ where NTHETA and NSIGMA are the number of THETA's and SIGMA's, respectively, in the model.

LTH should be set equal to the value of MAXPTHETA.

Edit resource\SIZES.f90 (Linux: resource/SIZES.f90) and supply the appropriate values . Retype the SETUP7 command, with options “same rec” instead of “big norec” or “reg norec”.

6) Whenever SIZES is changed it is necessary to recompile NONMEM using the rec option for the SETUP7 command. With Windows, SETUP7.bat may not warn the user that SIZES has been changed.

7) In some cases the \$COVARIANCE record may be incorrectly parsed resulting in an NM-TRAN error message. If this problem occurs, place the \$COVARIANCE record prior to the first \$ESTIMATION record in the control stream.

8) It has been reported that installation on Windows 7 using SETUP7.bat requires that the line

```
call .\util\dotest.bat %f%
```

be removed from SETUP7.bat. In addition, the User Account Control must be set to “do not notify” prior to installation. It can be turned back on after installation is complete.

9) With gfortran on Linux the SETUP7 script on the NONMEM 7.1.0 distribution CD does not provide optimization as the default or when it is requested. However, the SETUP7 script can be edited to provide an installation with optimization. (NOTE: The SETUP7.bat batch file for installation of NONMEM 7.1.0 on Windows does provide optimization as the default and if requested. No changes are needed to this file.)

Step 1) Copy the contents of the NONMEM 7.1.0 distribution CD to a directory, e.g. nm7CD, and change to that directory.

Step 2) Edit the SETUP7 script by adding the statements between #***'s to the section of SETUP7 code shown below.

```

# options for compile
op=" "
if [[ $o == y ]]; then
#*** Change IDS#1 30 Sept 2009
if [[ $ftail == gfortran ]]; then
op="-O3 -ffast-math "
fi
#***
if [[ $os == Darwin && $ftail == ifort ]]; then
op="-Gs -nologo -nbs -w -Ob1gyti -Qprec_div -4Yportlib -
traceback -vec-report0"
fi
if [[ $os != Darwin && $ftail == ifort ]]; then
op="-static -Gs -nologo -nbs -w -Ob1gyti -Qprec_div -
4Yportlib -traceback -vec-report0"
fi
if [[ $ftail == g95 && $os != "SunOS" ]]; then
op="-O3 -ffast-math -fsloppy-char"
fi
fi
fi

```

Step 3) Save the modified SETUP7 script.

Step 4) Make a new directory, e.g. ../nm7.

Step 5) Use the following or a similar command, depending upon other options that are chosen, to perform the installation.

```
/bin/bash SETUP7 <path>/nm7CD <path>/nm7 gfortran y ar
```

This will install into nm7 and preserve the nm7CD directory as an image of the CD but with the edited SETUP7 file.

Update to Known Problems

Correction to item 2 above and in the readme_7.1.0 file on the CD.

NONMEM 7.1.0 has been successfully installed on Vista 64-bit using gfortran v.4.5.0

Correction to Item 7 above and in the readme_7.1.0 file.

Another workaround is to include the option UNCOND (or COND, if that is your preference), e.g.

```
$COVAR PRINT=S UNCOND
```

The latest information about Known Problems will be posted to the nmusers listserve.

III. Compiler Related Information

gfortran: The name of the variable for the path to the lib directory differs across platforms. LD_LIBRARY_PATH is used for Linux and Unix. LIBRARY_PATH is used for Windows.

ar command: It may be necessary to provide a full path name for the ar command if multiple Fortran compilers are available.

ifort: for faster run times (as much as 8%) delete the /traceback option from the "set op" command of SETUP7.bat or delete the -traceback option from the "op=" command in SETUP7.

Then reinstall in the same directory. E.g.,

MS-DOS:

```
cd c:\nm7
SETUP7 c:\nm7 c:\nm7 ifort y link same rec
```

Linux:

```
cd /opt/nm7
/bin/bash SETUP7 /opt/nm7 /opt/nm7 ifort y ar same rec
```

Sun Studio 12 f95: The SETUP7 script included on the NONMEM 7.1.0 distribution CD does not provide for the installation on Solaris using the Sun Studio 12 f95 compiler.

However, there is now available an installation procedure for NONMEM 7.1.0 on Solaris 10 with the Sun Studio 12 f95 Fortran compiler. The necessary files and instructions can be obtained by sending an e-mail request to Thomas.Ludden@iconplc.com.

See the readme_7.1.0 file on the CD for additional information about compilers.

IV. Re-installing NONMEM 7 with changes

If NONMEM 7 has been installed in a given directory, it can be re-installed in a new directory. The CD is no longer necessary.

E.g.,

```
cd old-directory
SETUP7 old-directory new-directory ifort y link big rec
```

A different compiler, or different options (e.g. big vs. reg) may be specified. Option rec must be specified to recompile.

If, however, changes have been made to a file in a subdirectory (e.g., SIZES.f90 in directory resource), then do not use SETUP7 or SETUP7.bat in this way. All files in subdirectories will be replaced with files from nonmem7r.zip and nonmem7e.zip. Thus, the modified SIZES.f90 will be replaced by the original.

Instead, the operating system's copy command should be used to create the new directory. E.g.,

```
MS-DOS: xcopy /S /Q old-directory new-directory
Unix:    cp -R old-directory new-directory
```

Now re-install with option "same rec"

MS-DOS:

```
cd new-directory
SETUP7 new-directory new-directory ifort y link same rec
```

Unix:

```
cd new-directory
/bin/bash SETUP7 new-directory new-directory ifort y ar
same rec
```

V. Information Regarding Stable Model Development for Monte Carlo Methods

This section should be appended to "Intro to 7, I.30, Advanced Topics"

The Monte Carlo EM and Bayesian methods create samples of etas from multi-variate or normal or t distributions. Because of this, some extreme eta values may be randomly selected and sent to the user-developed model specified in \$PK, \$PRED, \$DES, and/or \$ERROR. Usually these extreme eta positions are rejected by the Monte Carlo algorithm because of the poor resulting objective function. But occasionally, floating point overflows, divide by zero, or domain errors may occur, which can result in failure of the analysis. This may occur especially when beginning an analysis at poor initial parameter values. At present, NONMEM is unable to recover from some of these errors, but in future versions we hope to provide more robust recovery algorithms. Here are some suggestions to provide a more robust user model that protects against domain errors or floating point overflows, or allows NONMEM to reject these positions of eta that cause them and continue the analysis.

If it is impossible to calculate the prediction due to the values of parameters (thetas or etas) from NONMEM, then the EXIT statement should be used to tell NONMEM that the parameters are inappropriate. The EXIT statement allows NONMEM to reject the present set of etas by setting an error condition index, which is in turn detected by classical NONMEM algorithms as well as the Monte Carlo algorithms. With the NOABORT switch of the \$EST statement set, NONMEM may then recover and continue the analysis.

For example, if you have an expression that uses

```
LOG(X)
```

You may wish to flag all non-positive values and let NONMEM know when the present eta values are unacceptable by inserting:

```
IF(X<=0.0) EXIT  
LOG(X)
```

On some occasions, you may need to have the calculations complete, then this expression could be transformed to:

```
LOG(ABS(X)+1.0E-300)
```

to avoid arguments to LOG that are non-positive.

If you have an expression which is ultimately exponentiated, then there is a potential for floating point overflow. An expression such as

```
EXP(X)
```

Which is likely to cause a floating point overflow could be filtered with

```
IF(X>100.0) EXIT  
EXP(X)
```

Again, if the calculation must complete, such as when evaluating a user-defined likelihood, then you can place a limiting value, taking care that it causes little first derivative discontinuity:

```
EXPP=THETA(4)+F*THETA(5)
```

```
;Put a limit on EXPP, as it will be exponentiated, to avoid floating
```

```

; overflow
IF(EXPP.GT.40.0) EXPP=40.0
F_FLAG=1 ; Categorical data
; IF EXPP>40, then A>1.0d+17, A/B approaches 1, 1/B approaches 0 and Y
; is approximately DV
A=DEXP(EXPP)
B=1+A
Y=DV*A/B+(1-DV)/B      ; a likelihood

```

If your code uses SQRT() phrases, the expression within parentheses should be always positive. Sometimes expressions are calculated to near zero but slightly negative values, such as

-1.1234444555E-16. Such values may legitimately be 0, but square rooting a negative number could result in failure of analysis. In such cases, the difficulty is due to the finite precision of the computer (e.g., rounding error causing a value to be negative that would be non-negative on a machine with infinite precision) then the code should be written so as to produce the correct result. To protect against this,

```
SQRT(X)
```

could be converted to

```
SQRT(ABS(X))
```

Or

```
SQRT(SQRT(X*X))
```

The EXIT statement should not be used in such near-zero cases. It could lead to a failure in NONMEM with a message containing text such as

```

DUE TO PROXIMITY OF NEXT ITERATION EST. TO A VALUE AT WHICH THE OBJ.
FUNC. IS INFINITE

```

An EXIT may still be issued for values of X that are clearly negative because of erroneous inputs, and you may wish to flag this calculation, so that the estimation algorithm rejects this position:

```
IF(X<=-1.0E-06) EXIT
```

```
SQRT(ABS(X))
```

Such protection codes described above need not be inserted for every LOG(), EXP, or SQRT, but only if your analysis fails frequently or tends to be sensitive to initial values.