

NONMEM V Supplemental Guide

March 2013

by

Alison J. Boeckmann

Stuart L. Beal

Lewis B. Sheiner

NONMEM Project Group

C255

University of California at San Francisco

San Francisco, CA 94143-0626

ICON Development Solutions, Hanover, Maryland

Copyright by the Regents of the University of California

1998,2013

All rights reserved

Preface

The appearance of the NONMEM V Supplement coincides with the appearance of the following programs:

NONMEM Version V Level 1.0

PREDPP Version IV Level 1.0


NM-TRAN Version III Level 1.0

They are referred to collectively as NONMEM V.

This Supplement describes changes found with NONMEM V of which the user should be aware, especially if he used earlier versions of NONMEM. It is accompanied by new versions of Users Guide Parts III (Installation), VI (Conditional Estimation Methods), and VIII (Help), which have been updated for the new release. New versions of the other guides will be distributed at later dates.

The material is roughly divided into separate chapters for NONMEM, PREDPP and NM-TRAN. For the most part, descriptions are given from the point of view of a user of NM-TRAN. Thus, a user who creates NONMEM control streams without using NM-TRAN will have to wait till the release of the updated version of Part I to learn how to specify new features to NONMEM. Most users use NM-TRAN exclusively, and perhaps they should read the NM-TRAN chapter first. Whenever possible, the user is referred to a guide that has been updated, such as Guide VIII, for further details about a given change.

With very few exceptions, NONMEM V is upwards compatible with earlier versions. However, the size of the NONMEM V executable file will be considerably larger than that of the corresponding NONMEM IV executable; see Users Guide Part III (Installation) Chapters I and III.

The symbol  in the right margin marks changes to NONMEM that may cause old NM-TRAN and NONMEM control streams to produce different results. It also marks changes that must be made to old PRED, PK, ERROR, DES, and AES user-supplied subroutines in order to use them with NONMEM V.

Note that no changes are needed to old \$PRED, \$PK, \$ERROR, \$DES or \$AES abbreviated codes or to old data sets.

The appearance of the March 2013 edition coincides with the appearance of NONMEM 7.3. The Guide has been reformatted. No changes have been made to the content.

Chapter 1 - NONMEM

Chapter 1 - NONMEM

General

1. More Etas, Epsilons, Thetas

The maximum number of θ 's allowable is set at 40.

The maximum number of η 's + ε 's allowable is set at 30.

The maximum number of parameters that may be estimated (not fixed in some way) is set to 70.

The maximum number of η 's allowable when the Laplacian estimation method is used with analytic second derivatives is set to 10.

These maxima may be increased.

See Guide III III.2.9.1

Changes to NONMEM Inputs

2. Band Symmetric Matrices

An initial estimate of a diagonal block of either the OMEGA or SIGMA matrices may have a band symmetric form, in which case the final estimate has the same form.

3. Superproblems

A superproblem is a sequence of problems within a NONMEM run that can be iterated. With every iteration each problem is run exactly as was specified for the first iteration. Differences in the results from a given problem between iterations are due to differences to input files that have been made by later problems within the superproblem or, when the problem involves simulation, to different random numbers being used. There can be a one-level nesting of superproblems, one superproblem within another.



With NM-TRAN a superproblem is specified by including the record

```
$SUPERPROBLEM SCOPE=n ITERATION=m
```

before the first \$PROBLEM record of the superproblem.

See Guide VIII \$super




4. COVARIANCE Record

Field 1 of the NONMEM COVARIANCE record may no longer have the value 1 (Covariance Step unconditionally implemented). It may have only values 0 (Conditionally implemented) or 2 (omitted).  

The option UNCONDITIONAL of the NM-TRAN \$COVARIANCE record is obsolete and is ignored; the Covariance Step cannot be implemented if the Estimation Step terminates unsuccessfully.

See Guide VIII \$covariance

5. TABLE Record

Options PRINT, NOPRINT, HEADER, NOHEADER, and FILE of the NM-TRAN \$TABLE record apply to the table defined by the record, and only to this table. They must be re-specified for each table to which they apply.   

Tables may be written to the same or to different table files.

The list of labels may be as long as 50 with the NOPRINT option.

See Guide VIII \$table

6. Large Data Files

NONMEM no longer needs a FINISH record when there are 10,000 or more records and when the NONMEM data set coincides with the NM-TRAN data set. If there is no FINISH record, the data are read to end-of-file.

7. Marginal (MRG_) Data Item

The marginal data item is a new NONMEM data item type. It identifies records for which NONMEM computes and displays marginal quantities (expectations). MRG_ is a reserved label for this data item.

See Guide VIII **mr**g , **expectation**

8. Raw-Data (RAW_) Data Item

The raw-data data item is a new NONMEM data item type. It identifies template records for which NONMEM computes and displays raw-data averages. RAW_ is a reserved label for this data item.

See Guide VIII **raw** , **data_average_block** , **data_average_example**

9. Model Specification Files

When intermediate output is requested in the Estimation Step, and a Model Specification File is also requested, NONMEM updates the MSF with every iteration summary. Thus, a run can be efficiently continued after it has been interrupted.

With previous versions of NONMEM it was not possible to input a MSF to repeat the Estimation Step using a method other than the one used to write the MSF. This is possible with NONMEM V, as long as the MSF contains the results of a search that terminated successfully.



Changes to NONMEM Output Files

10. New Output File INTER

It is now possible to see the iteration estimates in the original parameterization for those iterations whose summaries appear in intermediate printout, providing a MSFO is used. The estimates may be found in file INTER, which will exist only during the duration of the run.

See Guide VIII **\$estimation** , **monitoring of search**

11. NONMEM Report

Minor changes have been made to the wording of certain lines in the NONMEM and PREDPP output. In addition, it is possible that the results of the Estimation and Covariance Steps may be slightly different than with earlier versions of NONMEM.  

When a population conditional estimation method is used, for each element of η , the average conditional estimate of this element across individuals conditional on the final population parameter estimates ($\bar{\eta}$) is output (**ETABAR**), along with a P-value (**P VAL.**) indicating how far the average is from 0.

The labels for η 's appearing in the output are ETA1, ETA2, ... ETA9, ET10, ET11, ... ET70 and ETA(1), ETA(2), etc.

Output from the Covariance Step will look the same as before, except that when the sizes of the arrays that are printed exceed the largest size that can occur with NONMEM version IV (i.e. 75x75), an alternative form of output for these arrays is used. This is a compressed form that may be advantageous to view, even if the sizes of the array do not exceed 75x75. With this form, all zero values due to a parameter being fixed in some way are not printed. Compressed format is obtainable for arrays of all sizes by including the option COMPRESS on the \$COVARIANCE record.

See Guide VIII **\$covariance**

The range of data records contributing to a scatterplot is printed in the header of the scatterplot when the user has explicitly specified this range or when the range doesn't include all the records that one might want

to examine.

Simulated η values are displayable in tables and scatterplots provided that the ETA array is defined in common NMPRD7 (which is automatic when abbreviated code is used) and provided the option ONLYSIMULATION appears on the \$SIMULATION record.

See Guide VIII **nmprd7**

Changes to User-Written Code

12. PRED Routine Declarations

User-written PRED routines for population data that worked with previous versions of NONMEM will not work with NONMEM V until the declarations in these routines have been revised to reflect the larger numbers of η 's and ε 's allowed with NONMEM V.

(No change is needed to an old \$PRED block, however.)

See Guide VIII **pred**

13. ICALL

ICALL is an input argument of PRED, PK and other user-written routines. ICALL may now take values 5 and 6. These occur when NONMEM is computing expectations and raw data averages, which are new features of NONMEM V.

See Guide VIII **data_average_block**, **mrg_**, **raw_**

14. NEWIND

NEWIND is an argument of PRED and INFN and is available to PK and ERROR routines. The values of NEWIND have changed slightly.

NEWIND=0: First record of the data set. THETA value may differ from value at last call with this record.

NEWIND=1: First record of the data set, THETA value does not differ from value at last call with this record, and PRED is nonrecursive (with previous versions of NONMEM, this record would have had NEWIND=0).

NEWIND=1: First record of a subsequent individual record.

NEWIND=2: Subsequent data record of an individual record.

See Guide VIII **pred**, **\$pred**, **procm1**

15. F

With the Simulation Step, PRED may return the simulated observation as the DV data item, rather than in the argument F. With odd-type data the simulated observation must be returned as the DV data item.

See below 23 (Likelihood)

16. NONMEM Read-Only Commons

There are new NONMEM read-only commons and changes to certain existing commons (in particular, ROCM6).

See Guide VIII **NONMEM commons** as well as separate entries for each common.

17. NONMEM-PRED Commons

There are new NONMEM-PRED commons and changes to certain existing commons.

See Guide VIII **NONMEM commons** as well as separate entries for each common.

User-written PRED routines that are recursive should signal this fact to NONMEM using COMMON NMPRD8.

18. The PASS Routine

PASS now stores the data record as PASSRC in common NMPRD9, as well as in PRED's argument DATREC. It stores the NEWIND value as NEWND in common ROCM34, as well as in PRED's argument NEWIND.

When PASS is used, any PRED-defined items that are displayed are now available in common NMPRD4 with each data record. If one is using \$PRED, one may simply refer to these items by name as one is "passing through the data. If one is using PREDPP and an INFN routine, and the items are computed in \$PK or \$ERROR, then one should store the items in the reserved region of NMPRD4 so that, if the INFN uses PASS, the INFN routine can know where to refer to these items.

See Guide VIII **rocm34** , **nmprd9**

19. The MIX Routine

A MIX routine need only set NSPOP once, at the problem initialization call (ICALL=1). This number may no longer change between individuals. However, if for a given individual, the second (for example) subpopulation doesn't apply, then set P(2)=0 when MIX is called for that individual.

With earlier versions of NONMEM, a displayed value of an η was obtained under the subpopulation estimated as most probable. On the other hand, the displayed value of a PRED-defined variable was obtained using the estimate of η obtained under the highest-numbered of the subpopulations. With NONMEM V, the displayed value is obtained using the displayed value of η (i.e., under the most probable subpopulation). Values of the PRED-defined variable obtained using estimates of η under different subpopulations may be obtained and displayed as usual (by storing these values in COM variables allocated from the save region of common NMPRD4).

With the FO method, it has been possible to use a common η with two submodels. When conditional estimates are used, this has heretofore not been possible. Now it is possible. The conditional estimates of a common η will differ between the two submodels.

See Guide VII

New Methods of Estimation

20. METHOD=HYBRID ZERO=list

The Hybrid method is a hybrid between the FO and FOCE methods. The first-order expansion in η is made about a vector some of whose elements are 0 and others of which are a type of conditional estimate.

This method may be specified with \$NM-TRAN, using the options METHOD=HYBRID and ZERO=list on the \$ESTIMATION record.

See Guide VII

See Guide VIII **\$estimation**

21. CENTERING, FO

A conditional estimation method may use centering, which constrains the average conditional estimate of η to be nearly 0. This is useful when the amount of data per individual is large and when there is a large degree of model misspecification. A better fit (albeit perhaps not a very good fit, as the model is misspecified) can result. This method should not be used during model development.

This method may be specified with \$NM-TRAN, using the options METHOD=COND and CENTERING on the \$ESTIMATION record. The option LAPLACIAN can also be used.

With centering, the First-Order model, obtained by using the first-order expansion in η about the zero vector, may be used.

This method may be specified with \$NM-TRAN, using the options METHOD=COND, CENTERING, and FO on the \$ESTIMATION record. The option LAPLACIAN cannot be used (because the model is linear in η).

See Guide VII

See Guide VIII **estimation**

22. NUMERICAL

The Laplacian method may be implemented using numerical second derivatives. Use of analytical second derivatives is preferable, but there are situations where it may be useful to try using numerical second derivatives. Use of numerical derivatives may result in significantly longer computation time.

Numerical second derivatives may be specified with \$NM-TRAN, using the options METHOD=COND, LAPLACIAN, and NUMERICAL on the \$ESTIMATION record.

See Guide VII

See Guide VIII **\$estimation**

23. LIKELIHOOD, -2LOGLIKELIHOOD

By default, objective functions used with NONMEM are least-squares type functions, and as such, need not be considered to be related to likelihood functions. However, NONMEM allows data to be analyzed with an objective function equal to $-2\log$ likelihood for the data, and this is particularly necessary when the intraindividual distributions of the data are discrete and/or very asymmetrical. (For the purposes of this discussion, such data are called "odd-type" data.) To accomplish this, the value of F returned by the PRED routine with an observation may be set to the value of the conditional likelihood of η for the observation, given the values of the model parameters θ , rather than set to a prediction for the observation. The matrices G and H should be the usual derivatives of the expression used for F.

This different use of F is signaled by using the option LIKELIHOOD on the \$ESTIMATION record. It is assumed that first-order type approximations are not being used, and so the LAPLACIAN option should also appear.

Alternatively, F may be set to the value of $-2\log$ conditional likelihood of η for the observation. This is signaled by using the option -2LOGLIKELIHOOD (or simply -2LL) in the \$ESTIMATION record.

Odd-type data may also be simulated. For this purpose, PRED must return the value of a simulated observation as the DV data item (rather than in the argument F, as with standard-type data), a technique that will work even when the data are of standard-type.

See below 59 (Generated ID Data Items)

See Guide VIII **\$estimation** , **logistic_regression_example**

24. CONTR, CCONTR

With earlier versions of NONMEM, odd-type data have been analyzed using special routines CONTR and CCONTR, which are not supported by the NONMEM Project Group. All versions of CONTR and CCONTR that have been used for this purpose with earlier versions of NONMEM should not be used with version V. The option LIKELIHOOD should be used instead.






Chapter 2 - PREDPP

Chapter 2 - PREDPP

PK, TRANS, and ERROR routines

25. PK, TRANS, and ERROR Routine Declarations

PK, TRANS, and ERROR user-written subroutines for population data that worked with previous versions of NONMEM will not work with NONMEM V until the declarations in these routines have been revised to reflect the larger numbers of η 's and ε 's allowed with NONMEM V. (No change is needed to abbreviated code in an old \$PK or \$ERROR block, however.)   

See Guide VIII **pk** , **error** , **trans**


26. More Parameters

The maximum number of rows in GG (i.e., of basic and additional PK parameters) increases from 30 to 50. If necessary, this maximum can be increased even further.

See Guide III IV.1.3 V.2.6

See Guide VIII **pk** , **psizes** , **tsizes**

27. Common for DOSTIM

Variable DOSTIM and its η derivatives are defined in read-only common PROCM2. (In the documentation for PREDPP III, the variable was called DOST. The name has been changed to DOSTIM for documentation purposes. This does not require any change to existing PK routines.) The declaration of common PROCM2 has changed to reflect the larger number of η 's allowed with NONMEM V. 



See Guide VIII **procm2** , **\$pk**

28. Active Etas

A user written PK, TRANS, or ERROR routine may compute derivatives for all η 's, or optionally, to reduce run time, only for active η 's.

See Guide VIII **procm5**

29. Common PRRAND

With PREDPP Version III, realizations of the η and ε variables in the ERROR and PK routines may have been stored in a PREDPP common PRRAND. With PREDPP Version IV, η 's may be stored in a new NONMEM common, NMPRD7. This allows simulated η 's to be automatically displayable by NONMEM. ERROR may store ε 's in common PRRAND, but the declaration is different than it was previously. Declarations for PRRAND in old user-written PK and ERROR routines should be modified to include both NMPRD7 and PRRAND.  

See Guide VIII **nmprd7** , **prrand**

ERROR

30. ERROR's IDEF

At ICALL=1, a value may optionally be stored in IDEF(3) to reduce run time. No existing ERROR routine need be changed.

See Guide VIII **error**

Event Records

31. The Dose Rate (RATE) Data Item

Documentation change. Guide VI (pre 1998 editions) states: "A 0 indicates that the dose is an instantaneous bolus dose." There is an exception: a record with AMT=0, RATE=0, SS>0, and II=0 allows steady-state amounts to be computed by PREDPP based on system input given by the DES routine. This is useful when the differential equations provide for endogenous drug production. Furthermore, with NONMEM V, NM-TRAN supplies EVID=1 (dose event) for such records. Note that with NONMEM IV, NM-TRAN supplied EVID=0 (observation event) for such records, which is not correct. Consequently, if such records were present in the data set, the user had to provide the EVID item.

See Guide VIII **ss dose event record**

32. The Steady State (SS) Data Item

The SS data item may take the value 3. It is useful only with SS6 and SS9.

See Guide VIII **ss data item**

Other User Subroutines**INFN****33. Use of an INFN Argument**

At ICALL=0, INFN may now change the value of the first argument from 0 to a number in the range 1-8999. PREDPP prints this number in the output to help the user identify his own code. No existing INFN routine need be changed.

See Guide VIII **infn**

MODEL**34. More Compartments and More Parameters**

The maximum number of user-defined compartments with a general ADVAN routine increases from 9 to 19. If necessary, these maxima can be increased even further.

See Guide III IV.1.3, V.1.6 IV.1.3 , V.1.6

See Guide VIII **pk , psizes , tsizes**

35. Models with No Basic PK parameters

With ADVAN routines 6, 8, and 9, MODEL may now define models with no basic PK parameters.

See below 40. (Theta and EVTREC are Usable Directly)

DES**36. DES Routine Declarations**

A new argument DT must be declared. The partial derivative of DADT(i) with respect to T must be stored in DT(i). Old DES routines may or may not require a change. If T does not appear on the right side of any differential equation, then all elements of DT are zero and DES need not compute them because they are initialized to zero by PREDPP. If in addition the Fortran compiler and/or system loader does not consider it to be a fatal error if a calling routine passes more arguments to a subroutine than the subroutine declares, then old DES routines will still work. With some platforms, e.g., Microsoft, there will be an unresolved external reference, and it will be necessary to modify existing user-written DES routines to declare the new argument DT.

See Guide VIII **des**

37. Structure for DA, DP, DT: Full vs. Compact

Arrays DA, DP, DT can have "full format, as has been used in the past. E.g., DA is defined as DA(10,10), where DA(i,j) is the partial derivative of DADT(i) with respect to A(j). Now, they may also have "compact" format, in which (e.g.) DA is defined to be in effect one-dimensional and only those derivatives that can be non-zero for the particular model have defined positions in the array and need be computed.

Existing DES routines can be used with no change (but see 36). However, compact arrays are required with the Laplacian method if second-partial derivatives are computed analytically. The use of "compact" format is optional with other methods. Note that, by default, NM-TRAN implements \$DES blocks using compact formats. More information about compact array formats may be obtained by consultation with the NON-MEM Project Group.

The option DES of the \$ABBREVIATED record specifies the format of the arrays returned by the DES routine: FULL or COMPACT.

See Guide VIII **\$abbreviated , des , prdde1**

38. DES Initialization at ICALL=1

PREDPP Version IV makes an initialization call to the DES routine. This call does not occur with earlier versions of PREDPP. A new common is defined in PREDPP for DES and AES initialization:

```
COMMON /PRDDE1/ ICALL, IDEFD( 2 ), IDEFA( 2 )
INTEGER ICALL, IDEFD, IDEFA
```

ICALL has the same value as PK and ERROR's ICALL argument. At ICALL=1, DES may optionally store values in IDEFD in order to improve run time. Values may be stored in DADT, etc., as at other calls, but these values are ignored.

No changes are required to any existing DES routine because of this change. However, at ICALL=1 a user-written DES routine that does not check ICALL will compute DADT, etc., using values that PK computed (at ICALL=1) with the first event record and compartment amounts that are 0. There is some risk of arithmetic difficulties if, for example, a compartment amount of 0 is raised to a power. If this occurs, the user should insert the following statement at the top of the routine:

```
IF ( ICALL.EQ.1 ) RETURN
```

See Guide VIII **des , prdde1**

39. Displayed DES-defined Items

DES-defined items may be displayed in tables or scatterplots and are computed at the event time in the data record. With previous versions of PREDPP, the displayed items might have been computed at a slightly different time and might have had slightly different values.

40. Theta and EVTREC are Usable Directly

New PREDPP read-only commons contain the current THETA vector and the current event record (the event record to which the system is being advanced). Thus these quantities can be directly referenced by DES (rather than obtained as PK parameters). Similarly, \$DES abbreviated code may now include direct reference to THETA elements and data record items. This may improve run time.

See Guide VIII **procm6 , procm7**

41. Use of DOSTIM and DOSREC in DES

PREDPP defines variables DOSTIM and DOSREC in commons PROCM2 and PROCM3. Their use in DES was not previously documented.

A DES routine may test DOSTIM and DOSREC in a logical condition. It may use them on the right-hand side of an assignment statement. If DOSTIM is a random variable, DOSTIM must not be used in DES or \$DES. However, DOSTIM may always be used in a \$PK block or PK routine to define a random variable which may be used in the DES routine.

See Guide VIII **des** , **procm2** , **procm3**

42. Time T in DES routines

With PREDPP Version III, if a rate parameter, duration parameter, or absorption lag time parameter is a random variable, then T may be used only in logical conditions.

This restriction is now removed. Time T may be used freely in DES routines and in \$DES abbreviated code.

AES

43. AES Initialization at ICALL=1

PREDPP Version IV makes an initialization call to the AES routine. This call does not occur with earlier versions of PREDPP. A new common is defined in PREDPP for DES and AES initialization:

```
COMMON /PRDDE1/ICALL, IDEFD(2), IDEFA(2)
INTEGER ICALL, IDEFD, IDEFA
```

ICALL has the same value as PK and ERROR's ICALL argument. At ICALL=1, AES may optionally store values in IDEFA in order to improve run time. Values may be stored in A, etc., as at other calls, but these values are ignored. At ICALL=1, INIT has the value 1.

No changes are required to any existing AES routine because of this change. However, at ICALL=1, a user-written AES routine that does not check ICALL will compute A, etc., using values that PK computed (at ICALL=1) with the first event record and compartment amounts that are 0. There is some risk of arithmetic difficulties if, for example, a compartment amount of 0 is raised to a power. If this occurs, the user should insert the following statement at the top of the routine:

```
IF (ICALL.EQ.1) RETURN
```

See Guide VIII **aes** , **prdde1**

44. Displayed AES-defined Items

AES-defined items may be displayed in tables or scatterplots and are computed at the event time in the data record. With previous versions of PREDPP, the displayed items might have been computed at a slightly different time and might have had slightly different values.

45. Theta and EVTREC are Usable Directly

New PREDPP read-only commons contain the current THETA vector and the current event record (the event record to which the system is being advanced). Thus these quantities can be directly referenced by AES (rather than obtained as PK parameters). Similarly, \$AES abbreviated code may now include direct reference to THETA elements and data record items. This may improve run time.

See Guide VIII **procm6** , **procm7**

46. Use of DOSTIM and DOSREC in AES

PREDPP defines variables DOSTIM and DOSREC in commons PROCM2 and PROCM3. Their use in AES was not previously documented.

A DES routine may test DOSTIM and DOSREC in a logical condition. It may use them on the right-hand side of an assignment statement. If DOSTIM is a random variable, DOSTIM must not be used in AES or \$AES. However, DOSTIM may always be used in a \$PK block or PK routine to define a random variable which may be used in the AES routine.

See Guide VIII **aes** , **procm2** , **procm3**

47. Time T in AES routines

With PREDPP Version III, if a rate parameter, duration parameter or absorption lag time parameter is a random variable, then T may appear explicitly in the algebraic equations, as long as no differential equation depends on the equilibrium compartment amounts.

With PREDPP Version IV, this restriction has been removed.

The PREDPP Library**48. Laplacian Method**

The analytic Laplacian method is supported with all ADVAN, TRANS and SS routines with these exceptions:

ADVAN5 ADVAN7 ADVAN11 ADVAN12 SS5 SS7 SS11 SS12

With the above routines, the the Laplacian method may be used, but only if the \$ESTIMATION record includes the option NUMERICAL.

(For ADVAN11 and SS11, and ADVAN12 and SS12, see below ADVAN11 , ADVAN12")

See Guide VIII **\$estimation**

49. ADVAN5

Substantial time savings are possible when the PK routine limits calls to once per individual record (CALLFL=1 or ONCE PER IR).

See Guide VIII **calling protocol**

50. ADVAN7

Substantial time savings are possible when the PK routine limits calls to once per individual record (CALLFL=1 or ONCE PER IR).

See Guide VIII **calling protocol**

There is new error message. An example of this message follows:

```
NUMERICAL DIFFICULTIES OBTAINING THE SOLUTION.  NON-REAL
EIGENVALUES.  PERHAPS ADVAN5 SHOULD BE USED FOR THIS PROBLEM.
"LARGEST" VALUE ALONG SUBDIAGONAL OF SCHUR MATRIX:  0.423E-01
(DIAG:  0.84E+00  0.84E+00)
```

If the maximum value printed is very small (e.g., E-15), then the computations will probably proceed correctly if the error message is suppressed; information about suppressing the message may be obtained by consultation with the NONMEM Project Group. If the maximum value is sizeable (e.g., E-01), then the model should be scrutinized and if there are no mistakes, ADVAN5 should be used.

51. ADVAN9 as an Algebraic Equation Solver

ADVAN9 allows the user to define a model consisting only of equilibrium compartments, in which case ADVAN9 becomes a purely algebraic equation solver. The quantities A(i) may be thought of as unknowns to be determined rather than as "compartment" amounts.

Several changes are associated with this use of ADVAN9:

Records with EVID=1 and EVID=4 (dose, reset and dose) may not be present in the data set because doses cannot be placed in equilibrium compartments. Similarly, data items related to doses (RATE, AMT, SS, IL, ADDL) may or may not be used in the data set. If used, they must contain null values.

The TIME data item is optional.

If TIME is used, ADVAN9 is called exactly as other ADVAN routines are called: when the value of TIME increases.

If TIME is not defined, the AES routine specifies the calling protocol for ADVAN9: call with every event record (the default) or once per individual record.

If calls to ADVAN9 are limited, the CALL data item may be defined in the data set and given the value 10 to force additional calls to ADVAN9.

No calls to the DES routine occur. The DES routine may be omitted if the linker permits an unresolved external reference, or may be a dummy routine (stub) that simply executes a return statement. With NM-TRAN, a \$DES block may not be present but NM-TRAN generates code for a dummy DES routine.

See Guide VIII **aes** , **prdde1** , **call data item**

52. ADVAN11

ADVAN11 implements the Three Compartment Linear Model. It uses a subroutine EXP3 from the PREDPP library. It may be used with TRANS1, TRANS4, and TRANS6.

See Guide VIII **advan11** , **trans4** , **trans6**

53. ADVAN12

ADVAN12 implements the Three Compartment Linear Model with First Order Absorption. It uses subroutines EXP3 and EXP4 from the PREDPP library. It may be used with TRANS1, TRANS4, and TRANS6.

See Guide VIII **advan12** , **trans4** , **trans6**

54. TRANS6

TRANS6 may be used with ADVAN3, ADVAN4, ADVAN11 and ADVAN12. The parameters are ALPHA, BETA and (with ADVAN11 and ADVAN12) GAMMA, plus rate constants from peripheral compartments to the central compartment.

See Guide VIII **trans6**

Chapter 3 - NM-TRAN

Chapter 3 - NM-TRAN

General

55. Warning Messages

NM-TRAN Version III generates various informational and warning messages. These messages are not fatal (NONMEM may still be run) and may be ignored. Warning messages can be suppressed or their numbers limited for a given run by use of a new NM-TRAN control record \$WARNINGS.

See Guide III V.2.1.4

See Guide VIII \$warning

56. Day-Time Translation: the Year 2000

All versions of NMTRAN process the year correctly when all four digits are coded. However, earlier versions of NM-TRAN than III assume that two digit years are in the 1900's. With NM-TRAN Version III, a constant LAST20 specifies the highest 2-digit year nn that is assumed to be in the 21st. century, i.e., to represent 20nn. By default, this constant has the value -1, and all 2-digit years remain in the 1900's. LAST20 can be changed at installation time. It can also be changed at run time using the LAST20 option of the \$DATA record. As an example, consider this fragment of data:

```
# TIME DROP=DATE
    0 12-31-99
    0 01-01-00
```

Earlier versions of NM-TRAN, and the present version with LAST20 set to -1, produce an error message:

```
(DATA ERROR)      RECORD      2, DATA ITEM  2: *****
ELAPSED TIME MAY NOT BE NEGATIVE.
```

With LAST20 set to 0 or any larger value, there is no error. The second year is understood to represent 2000. The elapsed time is 24 hours.

With earlier versions of NM-TRAN, and with LAST20 set to -1, the years "00" and "0" are not leap years. With LAST20 set to 0 or any larger value, they are understood to be 2000, which is a leap year. This changes the translation of dates that span February to March.

A warning message related to leap year can be controlled at installation time.

See Guide III V.2.1.5

See Guide VIII \$data

57. More Digits After Day-Time Translation

Previous versions of NM-TRAN provide four digits (xxxx.xx) for the TIME data item after day-time translation. This allows TIME to be at most approximately 416 days ($416 \times 24 = 9984$ hours). NM-TRAN III provides five digits (xxxxx.xx), which allows TIME to be at most approximately 4166 days. Furthermore, when the \$DATA record includes the option WIDE, six digits (xxxxxx.xx) are provided.

See Guide VIII \$data

58. TIME and II Data Items: Conversion to Days

The option TRANSLATE on the \$DATA record directs NM-TRAN to divide the contents of the TIME and/or II data items by 24, thus converting hours to days. The user must insure that the units of PK parameters such as CL or K are consistent with this.

See Guide VIII \$data

59. Generated ID Data Items

NM-TRAN infers that the data are population data when certain things are true of the control stream. Add to the list in Guide IV Chapter II section C.4.2 the following:

- 6 An \$ESTIMATION record includes the option LIKELIHOOD or -2LOGLIKELIHOOD, and an abbreviated code is present which uses η 's.
- 7 The \$ESTIMATION record includes the option LIKELIHOOD or -2LOGLIKELIHOOD, and an \$OMEGA record is used.

In these cases the user need not supply the L1 data item label on the \$INPUT record, although it is not incorrect to do so.

NM-TRAN generates one or the other of the following warning messages in all cases. This permits the user to verify that NM-TRAN has made the correct inference.

```
(WARNING  1 ) NM-TRAN INFERS THAT THE DATA ARE SINGLE-SUBJECT.
(WARNING  2 ) NM-TRAN INFERS THAT THE DATA ARE POPULATION.
```

Control Records


60. INCLUDE Records

Records whose first non-blank characters are INCLUDE may be present anywhere in the control stream. Such a record causes NM-TRAN to read records from a given file into the control stream.

See Guide VIII **include**

61. Filenames on Control Records

Filenames on NM-TRAN records may consist of as many characters as fit on a single line. A filename may not contain embedded spaces. If it contains commas, semicolons, or parentheses, or if it starts with an equal sign, then it must be surrounded by single quotes ' or double quotes

An option name may not be used as a filename unless it is surrounded by quotes. 

62. \$PROBLEM Record

The text following \$PROBLEM is optional.

See Guide VIII **\$problem**

63. \$DATA Record

When the format option is omitted, the maximum record size in the NM-TRAN data set is increased from 300 to 999 characters.

The option IGNORE allows records starting with any alphabetic characters (not just a single specified character) to be ignored. The character # is the default. Records in the NM-TRAN data set whose first character is # are ignored, even if the \$DATA record does not include the IGNORE option.

The option NOOPEN specifies that the data set is not to be opened by NM-TRAN. This is useful in a multi-problem control stream when the data set is created by an earlier problem.

See Guide VIII **\$data**

64. \$SUBROUTINES Record

With \$SUBROUTINES LIBRARY, different abbreviated code may be present in different problems of a multiple-problem run.

The name to which a subname is set may be a filename as described in 61 (Filenames on Control Records) above.

See Guide VIII **\$subroutine , subroutine**

65. \$THETA Record

A new option NOABORTFIRST applies to the Initial Estimates Step.

See Guide VIII **\$theta**

66. \$MSFI Record

The characters "=n" of the NPOPETAS are optional. The option may be coded POPETAS. This option can be used to inform NM-TRAN that the data are population data. This may be needed when the \$ESTIMATION record does not appear.

See Guide VIII **\$msfi**

67. \$ESTIMATION Record

New options of the \$ESTIMATION record are described in Chapter 1 of this guide.

68. \$SIMULATION Record

NM-TRAN provides an initial call to SIMETA and SIMEPS. Additional calls to SIMETA and SIMEPS can be specified explicitly in abbreviated code in simulation blocks.

Data can be transgenerated using abbreviated code in a simulation block.

With the option ONLYSIMULATION, simulated η values are displayable by using ETA labels in \$TABLE and \$SCATTERPLOTS provided that the ETA array is defined in common NMPRD7, as is automatically the case when \$PK, \$ERROR, and \$PRED abbreviated code is used.

Options REQUESTFIRST and REQUESTSECOND cause NONMEM to compute first and second partial eta-derivatives during the Simulation Step.

Option NOPREDICTION can be used with ONLYSIMULATION and indicates both that the simulated value is returned in the DV data item, and that etas (if any) are population etas. NOPREDICTION may be needed when the \$SIMULATION record appears, and the \$ESTIMATION record does not appear.

See Guide VIII **\$simulation , simulation block , nmprd7**

69. \$TABLE Record

In a multi-problem run, the list may include PRED-defined labels from a prior problem.

The characters ETA(A) and ETAA are no longer accepted as symbolizing ETA(10).

There are new options:

FIRSTONLY
FORWARD NOFORWARD
APPEND NOAPPEND

See Guide VIII **\$table**

70. \$SCATTERPLOT Record

There are new options:

FIRSTONLY
ABS0 NOABS0
NOORD0
OBSONLY

See Guide VIII **\$scatter**

71. \$MIX Record

The \$MIX record marks the beginning of abbreviated code for the MIX routine.

See Guide VIII **mix**

Abbreviated Code

72. New Features

FORTTRAN CALL, DO WHILE, ENDDO, and RETURN statements may be used under certain circumstances.



FORTTRAN functions SIN, COS, and ABS are allowed, and may have random variable arguments. Note that the partial derivative of $ABS(X)$ with respect to η is mathematically undefined at $X=0$. We are arbitrarily defining it to be $dX/d\eta$. If the prediction depends on X , X must always be either positive or negative.

Parentheses may be nested.

Parentheses may be used in arithmetic expressions that are part of conditional expressions, but not when the parentheses include .AND. or .OR..

Conditional statements may be nested, unless random variables appear anywhere in the inner or outer blocks.

FORTTRAN WRITE and PRINT statements may be used, with restrictions.

Variables in generated code that symbolize (first-, and second-) partial η -derivatives of random variables will have different names with NM-TRAN Version III than with Version II.  

See Guide VIII **write print** , **abbreviated**

73. Restrictions on the Use of Random Variables in Abbreviated Code

Random variables may be assigned using conditional statements. Old code (using indicator variables) is still valid.

Random variables may be redefined, and may appear on the right hand side of conditional assignment statements. They may be used recursively in their own redefinition.

If a random variable is defined by "incomplete" conditionals (i.e., conditionals which do not include both true and false cases), and all tests fail, the random variable has the value zero. This is a major difference between random and non-random variables. Consider these examples:

```
IF (WT.GT.0) K=WT*THETA(1)*EXP(ETA(1))
IF (WT.GT.0) TVV=WT*THETA(1)
```

If $WT \leq 0$, K is zero.

If $WT \leq 0$, TVV retains its prior value, which may have been computed with a previous event record.

When a $COM(i)$ variable is defined by incomplete conditional statements and all tests fail, the variable retains its value (as do non-random variables).

See Guide VIII **abbreviated**

74. Improved Computation of Powers

Although when $B=0$ and $P>0$ the computation $R=B**P$ is mathematically defined and is 0, some floating point processors fail with a floating point error. NM-TRAN now causes powers to be computed, in both generated and Library codes, in a manner that avoids floating point errors in this circumstance. In effect, powers are now computed as in a work-around that has been suggested to numerous NONMEM users who have asked for help with this difficulty:

```
Q=0
IF (B.EQ.0) Q=1
R=(1-Q)*(B+Q)**P
```

If the work-around is already part of the abbreviated code it may be removed, but there is no harm if it remains; the generated or Library code will execute correctly, albeit more slowly than necessary.

75. Right-Hand Quantities

The reserved variable `Y` may now appear on the right-hand side in `$PRED` and `$ERROR` abbreviated code, e.g., `LOGY=LOG(Y)`.

See Guide VIII **\$pred** , **\$pred**

The variable `MIXP` symbolizes a mixture probability computed by subroutine `MIX`.

See Guide VIII **mix** , **mixp** , **rocm25**

Right-hand quantities of interest with simulation are `NREP` and `IREP`. Right hand-quantities of interest with multiple problem runs and with super problems are `NPROB`, `IPROB`, `S1NUM`, `S2NUM`, `S1NIT`, `S2NIT`, `S1IT`, and `S2IT`.

See Guide VIII **rocm10** , **rocm14**

If NM-TRAN appends the `EVID` and/or `MDV` data items, then `EVID` and `MDV` may appear on the right-hand side in `$PK`, `$ERROR`, `$DES`, and `$AES` abbreviated codes.

76. Pseudo-Statements

The statement

```
COMRES=-1
```

may optionally be enclosed within parentheses, and, if so, may optionally be coded on the first line of the block, e.g.

```
$PRED (COMRES=-1)
```

See Guide VIII **\$abbreviated**

77. ICALL Blocks

Abbreviated code between an `IF` statement that tests `ICALL` and the matching `ENDIF` is referred to as an `ICALL` block. E.g.,

```
IF (ICALL.EQ.4) THEN
... ICALL block ...
ENDIF
```

In general, there is no reason to include code testing for `ICALL.EQ.2` (regular call). Such code should normally not be in an `ICALL` block, as it is, in effect, the default code.

In this section, `ICALL` blocks that test for values other than 2 are discussed. Such blocks are named according to the value of `ICALL`:

- 0 Run initialization block
- 1 Problem initialization block
- 3 Problem finalization block
- 4 Simulation block
- 5 Expectation block
- 6 Data Average block

The code within an `ICALL` block may include nested `IF` statements but if so, any test of `ICALL` must be in the outermost `IF` statement.

In an `ICALL` block, nested `IF` statements may include random variables but eta-derivatives are not computed.

Certain FORTRAN statements are permitted in `ICALL` blocks.

Feature	ICALL Values	Kind of Record
CALL PASS (MODE)	0,1,3	\$PRED
CALL SUPP (ie , ic)	0,1,3	\$PRED
CALL RANDOM (n , R)	4,5	\$PRED,\$PK,\$ERROR
CALL SIMETA (ETA)	4	\$PRED,\$PK,\$ERROR
CALL SIMEPS (EPS)	4	\$PRED,\$ERROR
DO WHILE (condition) ... ENDDO	0,1,3,4,5,6	\$PRED,\$PK,\$ERROR†
RETURN	0,1,3,4,5,6	\$PRED,\$PK,\$ERROR†

ICALL values 0,1,3 are not appropriate for \$PK and \$ERROR.

ICALL value 6 is not appropriate for \$PK.

RETURN statements must be used with caution because their use may by-pass desired actions of the routine.

See Guide VIII **simulation , expectation , data average initialization-finalization , abbreviated**

NM-TRAN with PREDDP

78. \$BIND Record

The \$BIND record is permitted with user-written PK routines as well as with \$PK abbreviated code.

See Guide VIII **\$bind**

79. \$MODEL Record

The total number of compartments other than the output compartment defined on the \$MODEL record may be as large as 19. If necessary, this maximum can be increased even further.

See Guide III IV.1.3 , V.1.6

See Guide VIII **\$model , pk , psizes , tsizes**

The name of a compartment may not be identical to an attribute unless it is enclosed in quotes (' or "). E.g., COMP= (DEFOBS) is not permitted but COMP= ("DEFOBS , DEFOBS) is permitted.

When the number of compartments for ADVAN5 or ADVAN7 exceeds 10, the LINK clause syntax Knm=k may be ambiguous. The letter T may be used to separate the two compartment numbers KnTm.

See Guide VIII **advan5 advan7**

With ADVAN9, all compartments may have the attribute EQUILIBRIUM.

80. \$PK Record

When the number of compartments for ADVAN5 or ADVAN7 exceeds 10, the reserved variable Knm may be ambiguous. The letter T may be used to separate the two compartment numbers KnTm.

See Guide VIII **advan5 advan7**

The variable SC symbolizes the scaling parameter for the central compartment with general ADVAN routines (5 through 9) if there is a compartment whose name is CENTRAL.

Pseudo-statements defining the calling protocol (CALLFL=n) and the variable listing option (COMRES=n) may be coded within parentheses following the record name, or on the next line. Within the parentheses, the calling protocol may optionally be described by a descriptive calling protocol phrase. E.g., the following are equivalent:

\$PK (CALLFL=1)

\$PK (ONCE PER INDIVIDUAL RECORD)

See Guide VIII **calling protocol phrase , \$pk**

81. \$ERROR Record

Pseudo-statements defining the calling protocol (CALLFL=n) and the variable listing option (COMRES=n) may be coded within parentheses following the record name, or on the next line. Within the parentheses, the calling protocol may optionally be described by a descriptive calling protocol phrase. E.g., the following are equivalent:

```
$ERROR (CALLFL=1)
$ERROR (ONCE PER INDIVIDUAL RECORD)
```

See Guide VIII **calling protocol phrase, \$error**

82. \$DES Record

The allowable right-hand side quantities include the data items of a data record, symbolized by the labels and synonyms specified in the \$INPUT record, and values of the θ parameters, symbolized by THETA(1), THETA(2), etc.

If quantities depending only on data items and θ parameters are computed directly in \$DES rather than \$PK, improvement in run time is possible.

Reserved variables DOSTIM and DOSREC may be used under certain circumstances.

See above 41 (Use of DOSTIM and DOSREC in DES)

ICALL blocks testing for ICALL values 4 and 5 are permitted. However, none of the statements listed in 77 (ICALL Blocks)" above may be used.

With ADVAN9, a \$DES block may not appear when there are only equilibrium compartments.

With ADVAN10 and steady-state doses with constant infusion doses, basic PK parameters KM and VM may be used in the \$DES abbreviated code.

See Guide VIII **\$des**

83. \$AESINITIAL Record

When there are only equilibrium compartments and there is no TIME data item, ADVAN9 is called by default with every event record. \$AESINITIAL may specify a calling protocol for ADVAN9. Pseudo-statements defining the calling protocol (CALLFL=n) and the variable listing option (COMRES=n) may be coded within parentheses following the record name, or on the next line. Within the parentheses, the calling protocol may optionally be described by a descriptive calling protocol phrase. E.g., the following are equivalent:

```
$AESINIT (CALLFL=1)
$AESINIT (ONCE PER INDIVIDUAL RECORD)
```

See Guide VIII **\$aesinitial , calling protocol phrase**

84. \$AES Record

The index i for algebraic expressions E(i) is optional.

The allowable right-hand side quantities include the data items of a data record, symbolized by the labels and synonyms specified in the \$INPUT record, and values of the θ parameters, symbolized by THETA(1), THETA(2), etc.

If quantities depending only on data items and θ parameters are computed directly in \$AES rather than \$PK, improvement in run time is possible.

Reserved variables DOSTIM and DOSREC may be used under certain circumstances.

See above 41 (Use of DOSTIM and DOSREC in DES)

ICALL blocks testing for ICALL values 4 and 5 are permitted. However, none of the statements listed in 77 (ICALL Blocks) above may be used.

See Guide VIII \$aes