

NONMEM Users Guide

Introduction to Version VI

March 2013

by

Stuart L. Beal

Edited by Thomas Ludden and Alison Boeckmann

GloboMax
ICON Development Solutions
Hanover, MD

Copyright by the Regents of the University of California
2008,2103
All rights reserved

I.1. Introduction

The revised Introduction to NONMEM Version VI (2008) accompanies NONMEM VI 2.0. The appearance of this March 2013 edition coincides with the appearance of NONMEM 7.3. The Guide has been reformatted. No changes have been made to the content. All Version VI 1.0 bugs identified as of the date of this document have been fixed. Version VI 2.0 is upwards compatible with 1.0, 1.1, 1.2 and 1.3 Changes to the October 2006 document (which accompanied NONMEM VI 1.0) are in boldface. These are:

1.5a. Changes to SIZES

1.6a. Updated version of NONMEM Users Guide III

1.6b. Changes to SETUP and CDSETUP6.BAT

1.6c. Changes to NMFE6 and NMFE6.BAT

1.6d. Html with hyperlinks

1.6e. nmsee6.for - New version of nmsee

2.1.4. Stieltjes Method is fully implemented

2.1.4a. Further Notes on Stieltjes

2.1.9a. Implementing PRIOR routines

2.1.12a. Elements of ROCM50 may be used in abbreviated code

2.1.14. Default Boundary Test is User-selectable

2.2.9. Increase in the number of records

2.2.10. Increase in the maximum number of function evaluations

2.2.11. Table: PRED, RES, and WRES with NOAPPEND; more items

2.2.12. Table and Scatter: more PRED-defined labels

3.5. PREDPP: New Feature: Initial Steady State

3.6. PREDPP: New in SIZES: MAXFCN

4.1.24. NM-TRAN datafiles: LAST20=50

4.1.25. NM-TRAN input files: 160 characters per line

4.1.26. NM-TRAN datafiles: Ill-formed files

4.1.27. NM-TRAN datafiles: Blank lines in files (BLANKOK)

4.1.28. NM-TRAN datafiles: Filenames

4.1.29. NM-TRAN datafiles: Tab and ^M

4.1.30. \$DATA record: RECORDS option

4.1.31. \$PRIOR record

4.1.32. \$ESTIM record: NOOMEGABOUNDTEST option

4.1.33. \$ESTIM record: MAXEVAL option

4.1.34. \$TABLE and \$SCATTER: G and H elements

4.1.35. Abbreviated code: PHI

4.1.36. Abbreviated code: ROCM50

4.2.4. \$MODEL and \$PK: I_SS option (Initial Steady State)

5.4 New Reference: Wang, Y.

Appendix - Changes to Control Record Formats

This document provides an introduction to NONMEM Version VI, indicating the new features and basic changes from Version V. Perhaps the most important aspects of Version VI, however, are:

- 1) all Version V bugs identified as of the date of this document have been fixed,
- 2) numerous small improvements to Version V have been made, such as messages,
- 3) the stability and speed of Version V computations have been significantly improved, and depending on the model, size of the data set and estimation method, the user should notice this.
- 4) Version VI is essentially upwards compatible with Version V. The formatting of the report file, however, is a little different, and there are some minor changes noted below in sections 2.2.5 and 4.1.2 and 4.1.15 and 4.1.23. Model specification files created by version V cannot be used with version VI and vice versa. There is no single-precision version of NONMEM VI.
- 5) There is now a single INCLUDE file named SIZES. This file contains all the constants previously contained in NSIZES, PSIZES, TSIZES and LSIZES. SIZES contains a new variable, MAXIDS, the maximum number of individuals in a data set. The default value is 1000.

5a) Changes to SIZES

MAXFCN has been moved from PREDPP code to SIZES and has been given a larger value.

See I.3.6 below.

The following have larger values: LVR2, LNP4, LSW1, MAXIDS, PIR, DIMTMP DIMCNS, DINEW, DIMRV, DIMPXS.

(See the SIZES Help Item)

- 6) NONMEM Users Guides I-VIII and NONMEM V Supplement are provided as .pdf files on the NONMEM VI CD. With the exception of Guide II, these .pdf files are searchable. At this time, Guides **I-II and IV-VII** have not been updated for NONMEM VI. However, Guide VIII, the Help Guide, is fully updated to reflect changes from NONMEM V to NONMEM VI.

Changes for 2.0 are marked with "|" in the right margin.

6a) NONMEM Users Guide III (Installation) has been updated for NONMEM VI 2.0

6b) Installation utilities SETUP (Unix) and CDSETUP6.BAT have been changed.

SETUP now performs the complete installation on a Unix system. E.g., it makes the NONMEM VI 2.0 directory, copies tar files from the cdrom, and extracts all files from them. Previously, these steps had to be performed by the user. Thus, SETUP can be used much like CDSETUP6.BAT.

Both SETUP and CDSETUP6.BAT have a new argument, "recompile (y/n)", and will detect if the SIZES file is newer than the binary files. If SIZES is modified after NONMEM has been installed, or if post-installation changes or patches are made, SETUP (or CDSETUP6) may be run with recompile set to y, and all binaries will be recompiled. (Previously, the user had to remove old binary files in order to recompile.)

6c) NONMEM execution tools NMFE6 (Unix) and NMFE6.BAT have been changed. NM-TRAN messages (e.g., warnings and other informational messages) are now saved in a file FMSG, and are copied to the NONMEM report file as well as being written to the console.

6d) Html with hyperlinks

With NONMEM VI 1.0, there is a directory html on the NONMEM distribution medium containing html versions of the NONMEM help items. With NONMEM VI 2.0, the html files also include hyperlinks between help items, and between help items and the NONMEM Users Guide pdf files, including the present guide.

6e) nmsee6.for

There is a new version of nmsee, nmsee6.for. It is very similar to nmsee5.for, but with a few new options and prefixes. It is now described in on-line help and Guide VIII. All files may be obtained via ftp from

ftp://ftp.globomaxnm.com/Public/nonmem/nmsee6/

Note that this utility is provided for the convenience of the user but is not supported by Icon Development Solutions.

7) Updated descriptions of the NONMEM Control Record Formats are provided in an appendix to this document. **Changes are in boldface.**

NONMEM has three major parts: NONMEM, PREDPP, and NM-TRAN. Strictly speaking, "Version V" refers to the version of NONMEM, where the releases of these parts are V, IV, and III, respectively, and "Version VI" refers to the version of NONMEM, where the releases of these parts are VI, V, and IV, respectively.

Brief descriptions of new features and basic changes from NONMEM Version V that relate to the NONMEM program itself are found in section 2; those that relate to PREDPP are found in section 3; those that relate to NM-TRAN are found in section 4.

Detailed descriptions may be found in the Help items (NONMEM Users Guide VIII). The Help items should also be consulted for some new useful examples. Improvements have been made to many of the older items, in particular NMPRD5 (auto-correlation). Some terminology has been changed in a few help items. What were previously referred to as "scaled, transformed parameters" or STP are now designated as "unconstrained parameters" or UCP.

I.2. NONMEM

I.2.1. Estimation

1. With the Hybrid Estimation method and an intraindividual mean-variance model (see NONMEM Users Guide VII), the η, ε -interaction is now allowed. This interaction applies to the elements of η for which conditional estimates are obtained. Simply include the INTERACTION option on the \$ESTIMATION record.

(See the \$ESTIMATION Help Item.).

2. With the LAPLACIAN Estimation method and a mean-variance model (see NONMEM Users Guide VII), the η, ε -interaction is now allowed. In this case the numerical method for obtaining second partial η -derivatives must be used, and therefore too, a "slow" minimization method will be used. Simply include the INTERACTION option on the \$ESTIMATION record.

(See the Help Item.)

3. Background: With the First-Order Estimation method, if the model is not already linear in ε , it is first approximated by a first-order Taylor expansion in ε about $\varepsilon = 0$. This approximate model is then approximated further by a first-order Taylor expansion in η about $\eta = 0$, followed by the η, ε - interaction term being dropped. Finally, the Extended Least Squares Estimation method (1) is used.

Now the interaction term may be retained, resulting in the First-Order Estimation method with η, ε -interaction. Simply include the INTERACTION option on the \$ESTIMATION record. The result is generally not very different from that obtained when the interaction term is dropped. Generally, a slightly less biased estimate of Σ is obtained, and when there are systematic interindividual differences in the magnitude of interindividual random variability itself, a slightly less biased estimate of Ω is obtained.

(See the \$ESTIMATION Help Item.)

4. A new population estimation method - called the Stieltjes Estimation method - **is available in NONMEM VI 2.0**. It is a two-stage method. A population conditional estimation method should be used to "quickly" obtain fairly reliable parameter estimates; this comprises the first stage. Include the METHOD=CONDITIONAL option on the \$ESTIMATION record. The first-stage estimates are used as initial estimates in the second stage, wherein another objective function is minimized to obtain a final set of parameter estimates. The second stage takes much more time than the first stage, and the resulting parameter estimates are at least as good as the first-stage estimates, perhaps better. In the second stage, conditional estimates of η are not used, but the functions that are minimized to obtain conditional estimates are still evaluated at many values of η . Generally though, sufficiently good estimates are obtained by using conditional estimation methods, or even the First-Order Estimation method, and one needs to use the Stieltjes Estimation method only with somewhat pathological data designs. Simply also include the STIELTJES option (along with an optional GRID option) on the \$ESTIMATION record.

4a. Further notes on Stieltjes for NONMEM VI 2.0.

A set of tentative population estimates are first obtained using some 1st- or 2nd-order method specified on the \$ESTIM record in the usual way. (However, the Hybrid method cannot be used). At this first stage, for each individual, a tentative value for the integral (i.e. an area) is obtained. Then numerical integration is used to obtain second-stage estimates. The integral will be exact in the case where the intraindividual model is of mean-variance kind and is linear in eta. A default grid on eta-space is used for numerical integration.

This is a description of how one might set one's own grid. A grid is specified in terms of ellipsoidal coordinates, with origin at eta=0. With these coordinates, the length of a point P is given in fractile units - the first-stage estimate of the fraction of area enclosing all points with same or less length as that of P. A direction is specified by m=n-1 angles, where n is the number of (active) etas. At the second stage, the fraction of area enclosing all points of length less than r0 is taken to be identical to that computed at the first-stage. To obtain the fraction of area enclosing points of length greater than r0, numerical integration is used. A grid is obtained by first taking the interval [r0,r1] of the length axis and dividing this interval into nr equal subintervals. For each such subinterval, there is a region enclosing all points of length lying in the subinterval. There is yet another region (the tail region) enclosing all points of length between r1 and .9999. Within each region, the integrand is evaluated at a number of points, distributed fairly uniformly throughout the region, but with approximately the same length. The number of points is taken to be (ns**m)*(2**n), where ns is some integer. Think of ns as the number of points in a single quadrant of a 2-dimensional ellipse in n-space, and ns**m as the number of points in a orthant of m-dimensional space. 2**n is the total number of orthants. To use a different grid, include the option GRID=(nr,ns,r0,r1) on the \$ESTIM record. Constraints on these numbers are nr<=100, ns<=9999, 0<r0<r1<1. nr and ns should be integers. If r1>.9999, there is no tail region. The present default values are: GRID=(3,1,.6,.9).

(See the \$ESTIMATION Help Item.)

5. With minimizations that terminate with the message "MINIMIZATION SUCCESSFUL" in the report file, a warning message will now sometimes also be output:

HOWEVER, PROBLEMS OCCURRED WITH THE MINIMIZATION.

REGARD THE RESULTS OF THE ESTIMATION STEP CAREFULLY, AND ACCEPT THEM ONLY AFTER CHECKING THAT THE COVARIANCE STEP PRODUCES REASONABLE OUTPUT.

This may happen when e.g. the final gradient vector is too large.

6. Along with the ETABAR statistic, its estimated standard error is now also output. As with Version V, the corresponding P-value that is output relates to the hypothesis that the true value of ETABAR is 0. With Version VI, this is now a default P-value, and one may obtain an alternative P-value. This is done by implementing a subsequent NONMEM problem which simulates a new data set under the model used with the current problem and obtains parameter estimates for these new data along with an ETABAR statistic, and by requesting that the P-value corresponding to this statistic relate to the hypothesis that the true value of ETABAR for the subsequent problem is the same as that with the original problem. This provides a type of check on the adequacy of the data-analytic model (but the necessity to satisfy this check, like all such checks, is subjective). To get this alternative P-value, generate a model specification file with the original problem, input this file with the subsequent problem, and include the new option ETABARCHECK on the \$ESTIMATION record of the subsequent problem.

(See the \$ESTIMATION and ETABAR Help Items.)

7. With Version V, there are situations where the computation of ETABAR with a mixture model was not sensible. Now multiple ETABAR vectors are output, one for each subpopulation. For each subpopulation, ETABAR is computed over all individuals who have been classified (via a standard Bayesian type computation) to be in the subpopulation. Due to possible misclassification, the default P-value may not be reliable, and it is not output. One may, however, use the alternative P-value described in point 6.

8. A simultaneous analysis of both regular-continuous and odd-type data can be implemented more easily than can be done with Version V. A flag F_FLAG can now be set in a NONMEM-PRED common NMPR17 (via NM-TRAN abbreviated code), which directs NONMEM to interpret the value set in Y (in NM-TRAN abbreviated code) or in F (in a user-supplied PRED or ERROR routine) as being either a "prediction" or a likelihood, on an observation-specific basis.

(See the NMPR17 Help Item.)

9. A frequency-prior (2) may be specified for data-analytic purposes, or to simulate parameter values (in a Simulation Step). Two NONMEM utility routines help in this regard. NWPRI allows the THETA vector to be constrained by a "normal shaped prior", and/or the OMEGA matrix to be constrained by an "inverse-Wishart shaped prior", where both priors are specified by the user. TNPRI allows all parameters to be constrained by a prior that arises automatically from a NONMEM analysis of a prior data set.

(See the PRIOR and NWPRI and TNPRI Help Items and the NWPRI and TNPRI examples.)

9a. Implementing PRIOR routines has been made easier by a new NM-TRAN control record, \$PRIOR, which may be used instead of a user-written PRIOR routine.

(See the \$PRIOR help item.)

10. With an intraindividual mean-variance model (see Guide VII), the conditional likelihood for an observation is "conditioned" on values of the population parameters. With Version VI, it may easily be conditioned further; the condition being that the observation is inside or outside a given one- or two-sided interval. This is useful if e.g. blood concentrations below a known quantification limit have been eliminated from the data set. In this case one may use a one-sided interval with lower boundary equal to the quantification limit. It is assumed that the observations of the mean-variance model are normally distributed. With an observation record simply set the boundaries of the interval, YLO and YUP, in the NONMEM-PRED common NMPR12 (via NM-TRAN abbreviated code).

(See the NMPR12 and ROCM44 Help Items, and the YLO example.)

11. With an intraindividual mean-variance model (see Guide VII), the possible values of an observation may be categorized into one- and/or two-sided intervals, and the conditional likelihood for an observation may be taken to be the probability that the observation is in the category into which it is observed to fall.

Categorization may be preceded by conditioning as described in point 10 above. Interval boundaries may be random variables. It is assumed that the observations of the mean-variance model are normally distributed, so that the probabilities for the categories are essentially specified by probits. So this feature allows a probit analysis of data that might otherwise be handled by a logit analysis. With an observation record, simply set the lower boundary, CTLO, and upper boundary, CTUP, (and their η -derivatives) of the observed interval in the NONMEM-PRED commons NMPR13 and NMPR14 (via NM-TRAN abbreviated code).

(See the NMPR13 and NMPR14 and ROCM45 Help Items.)

12. The individual contributions to the objective function value are now stored in ROCM50 along with the individual ID values.

(See the ROCM50 Help Item.)

12a. Elements of ROCM50 may be used in abbreviated code.

See ROCM50 below.

(See the ROCM50 Help item and ROCM50 example)

13. There is a new option on the \$ESTIMATION record. The objective function value and the gradients can be calculated after the individual contributions have been sorted from the lowest to highest absolute values. In some cases this may decrease rounding errors. Simply include the SORT option on the \$ESTIMATION record.

(See the \$ESTIMATION Help Item.) The prototype code for sorting the individual components of the objective function value and the gradients was provided by Dr. Mark Sale, Next Level Solutions, LLC.

14. Default Boundary Test is User-selectable.

With NONMEM VI 1.0, a new test (the Default Boundary Test) was introduced that can give rise to the error message

PARAMETER ESTIMATE IS NEAR ITS BOUNDARY THIS MUST BE ADDRESSED BEFORE
THE COVARIANCE STEP CAN BE IMPLEMENTED

With NONMEM VI 2.0, this Default Boundary Test is optional for OMEGA, SIGMA, and/or THETA.

(See the \$ESTIMATION Help Item.)

I.2.2. General

1. There are a number of new NONMEM read-only commons, some containing particularly useful variables.

ROCM32: the numbers of the "current" individual record (NIREC) and data record (NDREC) within the individual record

ROCM46: the number of individual records containing at least one observation (NINDR), and the numbers of the first (INDR1) and last (INDR2) such individual records

ROCM48: the total number of data records in the "current" individual record (LIREC)

ROCM49: PRED_, RES_ and WRES_, the PRED, RES and WRES data items, available at problem-finalization (ICALL=3) during any pass through the data set

All these variables are available via NM-TRAN abbreviated code.
(See the ROCM32 and ROCM46 and ROCM48 and ROCM49 Help Items.)

2. Data records are passed sequentially to the PRED routine; heretofore, always from the first data record of an individual record to the last record of the individual record. With Version VI, variables (RPTI, RPTO, RPTON and PRDFL) in NONMEM-PRED common NMPR10 can be set and/or tested (via NM-TRAN abbreviated code), thus allowing a subsequence of data records to be repeatedly passed to PRED multiple times before the next data record following the last record of the subsequence is passed. Subsequences can be nested. This "repetition feature" allows e.g. kinetics to be computed by a convolution integral. The user can exercise control over which pass through a sequence it is, during which the output from PRED with a given data record will be used by NONMEM. This "PRED control" allows e.g. PRED output with a given record to involve computations over subsequent, as well as prior, records.
(See the NMPR10 Help Item and the Repetition 1 and Repetition 2 examples.)

3. The maximum number of subpopulations allowed with a mixture model is now an installation parameter in SIZES. The default value is 4, the maximum number allowed with Version V, and still, a fairly good number to use. With most data sets it is hard to discern more than 4 subpopulations. On the other hand, a recent paper (3) describes a technique for dealing with discrete covariates with missing values, which could greatly increase the required number of mixture subpopulations. (See the SIZES Help Item.)

4. With a mixture model, mixture probabilities can be individual-specific, and so the MIX routine is called repeatedly - with each individual. Useful individual-specific information for setting the probabilities may be made available to MIX by means of the \$CONTR record. With Version VI, the data record in NONMEM read-only common ROCM31 serves to provide an additional way for individual-specific information to be made available (and so now serves an additional purpose to that of providing a template data record). With each individual, when MIX is called, the first data record of the individual record may be found in the common. (see Section 4.1.6 and the ROCM31 and \$CONTR and \$MIX Help Items.)

5. With Version V, when a model specification file (MSF) is input, and the Simulation Step is implemented along with other NONMEM steps, the parameter values used in the simulation (the "true values"), as well as the initial parameter values used in conjunction with the other steps, are the final parameter estimates included in the MSF. With Version VI, more flexibility is available with a new option - TRUE - on the \$SIMULATION record, which allows the old behavior, *but this behavior is not the default behavior*.
(See the \$SIMULATION Help Item.)

6. At problem-finalization, a flag (SKIP_) can now be set in NONMEM-PRED common NMPR15 (via NM-TRAN abbreviated code), which permits a problem (with subproblems), or a superproblem or superproblem iteration to be ended and for NONMEM to continue with the next problem, superproblem or superproblem iteration, respectively.
(See the NMPR15 Help Item.)

7. There are a number of new NONMEM-PREDPP commons, some of which are described above. Of particular additional interest:

NMPR16: parameter values produced during the Simulation Step when a frequency prior (see above) is used during that step

8. There is a new step, NONPARAMETRIC. NONMEM obtains either marginal cumulatives or conditional (non-parametric) estimates of etas.

(See the \$NONPARAMETRIC and \$SIMULATION and ROCM18 Help items, and the example.)

9. Increase in the number of records that may be specified

If the number of records in the input data set is not specified in the NONMEM control file, NONMEM reads the data set to a NONMEM FINISH record or to end of file. With all previous versions of NONMEM, the maximum number of records that could be specified in the NONMEM control file was 9999. When the NONMEM data set contained more than 9999 records, NM-TRAN put a FINISH record at the end (unless the WIDE option of \$DATA was used). A FINISH record can cause problems, e.g., if another program is later used to process the NONMEM data set FDATA. With NONMEM VI 2.0, the maximum number of records that may be specified is 99999999. Only if the number of records is larger than 99999999 will a FINISH record will be present in FDATA. (See the FINISH Help Item.)

10. Increase in the maximum number of function evaluations

Previously, the maximum number of function evaluations that could be specified for the the Estimation Step was 9999. Now, the maximum is 99999999. It may still be useful to specify a smaller number of function evaluations and use a Model Specification File to continue the Estimation step in a later run, but is it not necessary.

11. Table: PRED, RES, and WRES with NOAPPEND; more items

When the NOAPPEND option of a \$TABLE record is used, PRED, RES, and WRES may be listed anywhere in a table and will appear where they are listed. Previously, they could listed anywhere in the table but would appear in the NONMEM table only as the final items (unless NOAPPEND was specified, in which case they would not appear at all).

When the NOAPPEND option of a \$TABLE record is used, up to 54 items can be output in a table file. Otherwise, the limit is 50. Previously, these limits were 24 and 20, respectively.

12. Table and Scatter: more PRED-defined labels

Up to 50 distinct PRED-defined items may be displayed in all tables and scatters combined. Previously, the limit was 20.

I.3. PREDPP

1. There is a new feature: model event times. These are additional PK parameters MTIME(i) defined in the PK routine or \$PK block. A model event time, like an absorption lag time, defines a time to which the system is advanced, but whose value usually cannot be known in advance. When the time is reached, certain indicator variables are set and a call to PK is made. At this call PK, DES, AES and ERROR can use the indicator variables to change some aspect of the system, e.g., a term in a differential equation, or

the rate of an infusion. Thus, a model event time can be used e.g. to mark the time at which the gall bladder begins to empty. A model event time could be implemented by using an absorption lag time, and this is what has been done until now, but doing this is somewhat clumsy. There may be up to PCT model event times, where PCT is a new installation parameter defined in SIZES. Its default value is 30.

(See the MTIME and \$PK and PRDPK1 and PRDPK2 Help Items and the MODEL TIME Example)

2. There is a new variable A_0FLG - the "compartment initialization flag" - in a new PREDPP read-only common PROCMP. When event records are passed to routine PK, PREDPP sets A_0FLG to 1 with the first event record of an individual record or with any reset record, and to 0 with all other records. When A_0FLG=1, compartment amounts have been initialized to zero, i.e. the state vector A (and its η -derivatives) has been set to zero, but PK may reinitialize the compartment amounts to nonzero values by resetting the state vector (and its η -derivatives) to such values (in a new PREDPP-PK common PRDPK3).

(See the \$PK and PROCMP and PRDPK3 and COMPARTMENT INITIALIZATION BLOCK Help Items.)

A_0FLG is a reserved variable in \$PK. See section 4.2 below for further discussion as to how to initialize compartment amounts.

3. With the PK routine, the compartment amounts (and their η -derivatives) may be found in PREDPP read-only common PROCMP4. These are the latest computed compartment amounts. That is, they are the amounts at the previous event time, or if at a later time (but before the event time for which it may be that PK is being called) a lagged or additional dose was given, or a regular infusion was terminated, or an event occurred at a modeled event time, then they are the amounts at the latest such time. Now, with Version VI, there is a new PREDPP read-only common PROCMP9 in which the time at which these compartment amounts are in fact computed may be found.

(See the PROCMP4 and PROCMP9 Help Items.)

4. When the value of the compartment data item (CMT) is n+1, where n is the total number of compartments in the kinetic system, the data item designates the (special additional) output compartment. With Version VI, this can also be done with the value 100, which is useful when n changes from run to run.

(See the CMT Help Item.)

5. PREDPP: New Feature: Initial Steady State

With NONMEM VI 2.0, PREDPP may be requested to compute steady state as the initial condition for a model. This feature is for use with general non-linear models (ADVAN6, ADVAN8, ADVAN9), when endogenous doses are present in the differential equations. Previously, the only way to do this was by using a dose event record having SS>0, AMT=0, RATE=0. With this feature, the initial condition is computed prior to the first event record (and after a reset event record) exactly as if such a dose record were present. It is not necessary to include the a dose record or, indeed, to define the data item SS, AMT or RATE. There is a new common PRDPK4.

(See the PRDPK4 and ADVAN6 ADVAN8 and ADVAN9 and \$MODEL Help Items.)

6. PREDPP: New in SIZES: MAXFCN.

Routines ADVAN6, ADVAN8, ADVAN9, and SS6 use a variable MAXFCN to specify the maximum number of calls to FCN1 (ADVAN6, ADVAN8, SS6) or RES (ADVAN9) during an integration interval. Previously, this variable was set to 100000 in the routines themselves, and the user could supply a larger value by setting IMAX in PRCOMG (e.g., by using verbatim code in the \$PK

block). Now, MAXFCN is set in SIZES, and is given a larger value 1000000. If an even larger value is desired, SIZES may be changed and PREDPP recompiled, or IMAX in PRCOMG may be set to the larger value using verbatim code.

Old verbatim code that changes IMAX will work as it did with previous releases, but, if IMAX is being set to 1000000, the verbatim code is no longer needed.

(See the SIZES and PRCOMG Help Items.)

I.4. NM-TRAN

I.4.1. General

1. The IGNORE option on the \$DATA record has been expanded so that any data record will be dropped when the values of one or more data items meet certain user-specified criteria, and a new ACCEPT option has been introduced so that any data records will be dropped unless the values of one or more data items meet certain user-specified criteria.

(See the \$DATA Help Item.)

2. With Version V, the option $\text{TRANSLATE}=(\text{TIME}/24)$ (or $\text{TRANSLATE}=(\text{II}/24)$) on the \$DATA record results in the TIME (or II) data items in units of hours being converted to data items in units of days, with two figures to the right of the decimal point being retained. With version VI, this option results in three figures to the right of the decimal point being retained, as does the option $\text{TRANSLATE}=(\text{TIME}/24.000)$ (or $\text{TRANSLATE}=(\text{II}/24.000)$). The option $\text{TRANSLATE}=(\text{TIME}/24.00)$ (or $\text{TRANSLATE}=(\text{II}/24.00)$) results in two figures to the right of the decimal point being retained.

(See the \$DATA Help Item.)

3. With Version V, the DO WHILE construction is allowed in an initialization/finalization block, and when this is used in conjunction with direct calls to the NONMEM utility routine PASS, passes through the data set may be implemented during problem initialization/finalization. Now a new DO WHILE (DATA) construction is allowed, which automatically and transparently manages the calls to PASS.

(See the ABBREVIATED CODE and PASS Help Items.)

4. With Version V, a random variable defined in an abbreviated code for a given routine must be completely defined with each data record that is passed to the routine. NM-TRAN now supports recursive definition of random variables in \$PRED, \$PK and \$ERROR blocks. Examples of recursively defined random variables are given in the following fragments of code.

Example 1 - random variable retains value from previous data record.

This example illustrates how one can use abbreviated code when WT is included only on some (e.g. the first) data records of an individual record. (Of course, this can lead to problems with developing plots involving WT.) Here, K is defined in terms of ETA(1), and so it is a random variable.

```
IF (WT.GT.0) THEN
  K=THETA(1)*WT*EXP(ETA(1))
ELSE
  K=K
ENDIF
```

Example 2 - bolus dose at time 0

This example illustrates how one can use abbreviated code to implement recursive kinetics in \$PRED assuming that TIME is the elapsed time since the previous data record.

```
K=THETA(1)*EXP(ETA(1))
IF (TIME.EQ.0) OLDA=AMT
A=OLDA*EXP(-K*TIME)
OLDA=A
```

Example 3 - multiple bolus doses

This example further illustrates how one can use abbreviated code to implement recursive kinetics in \$PRED assuming that TIME is the elapsed time since the previous data record.

```
K=THETA(1)*EXP(ETA(1))
IF (TIME.EQ.0) THEN
  A=AMT
ELSE
  A=A*EXP(-K*TIME)+AMT
ENDIF
```

5. The variables mentioned in sections 2 and 3: F_FLAG, YLO, YUP, CTLO, CTUP, NIREC, NDREC, NINDR, INDR1, INDR2, LREC, PRED_, RES_, WRES_, RPTI, RPTO, RPTON, PRDFL, and SKIP_ are all available with abbreviated code, as described in those sections.

6. Other variables that now may be used (as right-hand quantities) in abbreviated code are:

NEWL2 (from common ROCM17)

TEMPLT(n) (from common ROCM31)

The TEMPLT(n) variables refer to the items of the template data record. They may be used in data average blocks (ICALL=6) and \$MIX blocks. The items of the template record may be referred to by position or by label, e.g., TEMPLT(1) or TEMPLT(ID).

(See ROCM17 and ROCM31 Help Items.)

7. With Version VI, the variables S1SC and S2SC of NONMEM read-only common ROCM14 are no longer recognized in initialization/finalization blocks; rather, the variables S1NUM and S2NUM are used. (See ROCM14 and \$SUPER Help Items.)

8. User-written functions FUNCA, FUNCB and FUNCC (called abbreviated functions) may now be used as right-hand quantities in abbreviated codes. Such a function may depend on arguments which may be random variables, in which case the function too becomes a random variable. Elements of vectors (VECTRA(n), VECTRB(n), VECTRC(n)) may be defined as left-side quantities in abbreviated code. A vector may be used as an argument to an abbreviated function. The code for the function itself must be written by the user in FORTRAN.

(See the ABBREVIATED FUNCTION Help Item.)

9. The functions TAN, ASIN (arcsin), ACOS (arccosine), ATAN (arctangent) may now be used in abbreviated code. The INT function may also now be used. When this function is used outside a simulation block, INT of a random variable is regarded as a non-random variable.

10. The NONMEM utility routine RANDOM may now be used in initialization/finalization blocks.

11. The value of the RECORDS option of the \$DATA record may now be a label (appearing in the \$INPT record) of a data item type. In particular, it may be ID (alternatively, IR, INDREC, or any initial substring of length 3 or more of INDIVIDUALRECORD). With this value NM-TRAN understands the data records for the problem to start with the first data record of the NM-TRAN data set (from the place where the NM-TRAN data set is positioned before data records are read; see the NOREWIND option) and to include all data records in the series of contiguous data records that follow and having the same value of the data item as does the first record. It counts the number of these data records (minus comment or dropped records) and puts this number in the NONMEM control file, so that these data records (or NM-

TRAN-modified versions of them) become the data records used in the problem.
(See the \$DATA Help Item.)

There are two obvious applications. First, during a checkout phase, one might want to see only the predictions for the first individual record, and therefore, use a data set comprised only of the data records therein. When RECORDS=ID is used, one need not count these data records. Second, one might want to analyze each subject's data independently of the others. One way this can be done is by creating a single control stream with as many problems as there are individual records in the data, and including one each of the following series of \$DATA records in the series of problem specifications:

```
$DATA filename RECS=n1  
$DATA filename RECS=n2 NOREWIND  
$DATA filename RECS=n3 NOREWIND  
etc.
```

where "nk" is the number of records in the kth individual record. It would be necessary to count the exact number of records. Now, these records can simply be coded

```
$DATA filename RECS=ID  
$DATA filename RECS=ID NOREWIND  
$DATA filename RECS=ID NOREWIND  
etc.
```

(See \$DATA Help Item.)

12. With the INCLUDE record, the filename may now be followed by an integer n, whose default value is 1. Then NM-TRAN reads n copies of the named file.

If in the example above, where the problem specification for all subjects after the first are completely identical, a compact way of writing the control stream is possible. The control stream would contain the problem specification for the first individual, including the \$SUBROUTINES record, abbreviated code, and the \$DATA record for the first subject, and would end with (if there are e.g. 12 subjects in all)

```
INCLUDE ctlfile2 11
```

The file ctlfile2 would contain one problem specification for one subject, with no \$SUBROUTINES record or abbreviated code, but would include

```
$DATA filename RECS=ID NOREWIND  
(See INCLUDE Help Item.)
```

13. The INTERACTION option may now be used on the \$ESTIMATION record along with METHOD=HYBRID, or METHOD=CONDITIONAL, LAPLACIAN, or METHOD=ZERO. The ETABARCHECK option may be used on the \$ESTIMATION record. (see section I.2.1).
(See \$ESTIMATION Help Item.)

14. The TRUE option may be used on the \$SIMULATION record (see section 2.2 and the \$SIMULATION Help Item).

15. With Version V, when the data are single-subject and the L2 data item is defined, NM-TRAN creates new data items and labels them .ID.. Now the ID data item is taken to be the L2 data item, and new .ID. items are not created. The ID item may be defined, but then it is not in turn identified to NONMEM as such. The labels L2 and ID (if defined) may be used in abbreviated code, and then they refer to the corresponding items of the data record, as specified in the \$INPUT record.
(See the L2 Help Item.)

However, if, as previously, the user uses the label L1 instead of the label ID, or uses the label L1 as a synonym with the ID label, then NM-TRAN does not change the designation: For NONMEM, the items labeled L1 are taken to be the ID data items.

(See the ID and L2 Help Items.)

16. OMEGA(i,j) and SIGMA(i,j), where the subscripts i and j are integer constants, can now be used as right-hand quantities in \$PRED, \$PK, \$ERROR and \$INFN abbreviated codes. For example, when PREDPP is used with a slope-intercept residual error model, the computation of individual weighted residuals (with F evaluated at $\eta=0$) could be done as follows:

```
$ERROR
Y=F+EPS( 1 )+F*EPS( 2 )
IF ( COMACT.EQ.1 ) THEN
    STD=SQRT( SIGMA( 1 )+F**2*SIGMA( 2 ) )
    IWRES=( DV-F )/STD
ENDIF
```

When the two subscripts i and j are the same, only one subscript is needed. The OMEGA and SIGMA elements may be listed in WRITE and PRINT statements in initialization and finalization blocks.

(See the ROCM22 and INITIALIZATION-FINALIZATION BLOCK and WRITE/PRINT Help Items.)

17. SETHET(i,j), SEOMEG(i,j) and SESIGM(i,j), where the subscripts are integer constants, can now be listed in WRITE or PRINT statements in finalization blocks. These are elements of the estimated standard error arrays for the final estimates of THETA, OMEGA and SIGMA.

(See ROCM7 Help Item.)

18. Each of the array names THETA, OMEGA, SIGMA, SETHET, SEOMEG, SESIGM, and ETA - *without subscripts*, may be listed in a WRITE or PRINT statement, and then the entire array is written. With a WRITE statement, the format should either be omitted or be "*"; NM-TRAN creates a suitable format. E.g., either of

```
WRITE ( 99 ) THETA,SETHET
WRITE ( 99 , * ) THETA,SETHET
```

may be used. In such a statement, only arrays may be listed, and only if these arrays share the same dimensions. The array names THETA and ETA may be listed within any block of abbreviated code in which the elements of these arrays may appear on the right (e.g. \$PRED and \$INFN). The array names OMEGA and SIGMA may be listed within any initialization/finalization block. The array names SETHET, SEOMEG and SESIGM may be listed within any finalization block.

With arrays OMEGA, SEOMEG, SIGMA and SESIGM, the option (BLOCK) or (DIAG) *must be given* to indicate if the full array or just the diagonal elements are to be written. If the (BLOCK) option is used, the array is displayed in full symmetric form. E.g.,

```
IF ( ICALL.EQ.3 ) WRITE ( 99 ) OMEGA(BLOCK)
```

may be used.

(See the ROCM7 and ROCM22 and INITIALIZATION-FINALIZATION BLOCK and WRITE/PRINT Help Items.)

19. Character constants may be included in WRITE and PRINT statements in abbreviated code. E.g.

```
PRINT * , 'CLEARANCE EQUALS: ', CLE
```

(See the WRITE/PRINT "Help Item

20. The statements OPEN(u), CLOSE(u), REWIND(u), where u is a I/O logical unit number, are now usable in abbreviated codes. The unit number must be an integer constant between 40 and 99. A file name may be included with the OPEN statement, i.e. OPEN(u,FILE=filename). With OPEN and CLOSE,

the NONMEM utility routine FILES is automatically called.

(See the Abbreviated Code and FILES Help Items.)

21. Copying blocks of abbreviated code begin with tests of the reserved variable COMACT, as follows:

```
IF (COMACT.EQ. . . . ) THEN
. . . .
ENDIF
```

Unlike with Version V, η -derivatives are no longer computed in a copying block. COM(i) variables that are used on the left in a copying block are referred to as "explicit" save variables. They are automatically stored in the SAVE region (a subregion of the reserved region of common NMPRD4). When COM(i) variables are defined in a copying block, the reserved region of common NMPRD4 must be explicitly set up with the use of the COMRES option on the \$ABBREV record, but the COMSAV option need not be used.

Variables other than COM(i) variables may be used on the left in a copying block, and they are referred to as "implicit" save variables. These may not be defined along with COM(i) variables. Implicit save variables too are automatically stored in the SAVE region, and in this case, a reserved region of common NMPRD4 is automatically created; neither the COMRES nor COMSAV options can be used on the \$ABBREV record. Use of implicit save variables can obviate the need for a \$ABBREV record.

(See the COM COMACT COMSAV COMRES Help Item.)

22. A new NM-TRAN record concerns the computation of raw data averages. Heretofore, a data item occurring in the data record after the DV data item was not used in template matching. Now, by default it is. If a data item is to be excluded from template matching, the label or synonym for this item should be included in the new record - \$OMIT. E.g.

```
$OMIT TIME
```

As always, data items required by NONMEM are automatically excluded from template matching. The label of such an item may be included in the \$OMIT record, but this is unnecessary.

(See the \$OMIT Help Item.)

23. With Version V, the OTHER option on the \$SUBROUTINES record could be used at most three times. Now it can be used at most forty times.

24. NM-TRAN datafiles: LAST20=50

With NONMEM VI 1.0, a change was made to constant LAST20 in NM-TRAN's ABLOCK. Previously, it had the value -1. Two-digit dates were assumed to be in the 1900's. Now, the default value is 50. Two digit years are interpreted by default as follows:

00-50 = 2000-2050

51-99 = 1951-1999

(See the \$DATA Help Item.)

25. NM-TRAN input files: 160 characters per line

With NONMEM VI 1.0, a change was made to NM-TRAN. Previously, it read only 80 characters per line of the NM-TRAN control stream. This could lead to errors when the NM-TRAN control

stream contained more than 80 characters in a line. Now, it reads 160 characters per line. However, generated code (including that generated for lines of verbatim code) still contains 72 characters per line.

NONMEM VI 2.0 Changes

26. NM-TRAN datafiles - Ill-formed files

It is possible to create a file whose last line contains non-blank characters, but which does not end with carriage return/line feed characters (or, in the Unix environment, a newline character). This can be done using some editors (e.g., Wordpad, Notepad, Staroffice). Such a line is referred to here as an ill-formed line. What happens when an ill-formed line is read by the Fortran I/O library routines is not specified by the Fortran 77 standard. Compilers differ. Here are a few examples.

(1) gfortran - GNU Fortran 95 compile on Linux. f90: WorkShop Compilers 5.0 98/12/21 FORTRAN 90 2.0 on Solaris.

These compilers take the correct action: the characters on the last line are read normally, and the subsequent read command gets an End-of-file indication. In effect, the Fortran I/O routines have inserted the missing newline.

(2) g77 - GNU Fortran for Windows. f77: WorkShop Compilers 5.0 98/12/15 FORTRAN 77 5.0 on Solaris.

An End-of-file occurs when the last line is read, but the characters on the last line are actually read prior to the end-of-file condition occurring.

(3) It is possible that some compilers might ignore the characters on the ill-formed line entirely, and produce an EOF condition without first transferring any data. It is not known if there are any such compilers.

With compilers such as (2) and (3), and all previous versions of NM-TRAN, the characters on the last line are ignored. A data file that ends with an ill-formed line will, in effect, contain one line fewer than expected. If a control file ends with an ill-formed line, any task or options specified on that line will not be performed.

NM-TRAN has been modified so that, with compilers such as (2), it now detects when non-blank characters are read prior to an EOF, and generates an error message 589 (for a control or include file) or 590 (for a data file). The user should repair the file.

Note that with compilers (1), NM-TRAN continues to work correctly. It will see the last line, as before, and will not produce an error message.

27. NM-TRAN datafiles - Blank lines in files (BLANKOK)

Blank lines in the NM-TRAN data set are no longer permitted by NM-TRAN. A new option of the \$DATA record, BLANKOK, restores the previous behavior.

(See the \$DATA Help Item.)

28. NM-TRAN datafiles - Filenames

Filenames on NM-TRAN control records may now contain embedded spaces if the file is opened by NM-TRAN (rather than NONMEM). Filenames are now restricted to the number of characters that can be used by NM-TRAN (80) or NONMEM (71). Previously, embedded spaces were not permitted, and file names larger than the maximum would be truncated.

(See the \$DATA Help Item.)

29. NM-TRAN datafiles - Tab and ^M

Previously, NMTRAN did not allow a character in the control stream or the input data set that is less than the space character in the computer's collating sequence ("low-value character"). In particular, this meant that tab (ASCII 011) and carriage return (ASCII 015) characters were not permitted. (An exception is a data set for which a FORTRAN format specification is provided on the \$DATA record; NMTRAN did not detect low-value characters. The result is platform-dependent, i.e., it depends on the action taken by the Fortran library I/O subroutines.)

Tab characters can arise with text editors that save tab as an explicit character, rather than expanding it to spaces. They are also present in data that is cut and pasted from Excel to an ASCII file, or saved directly from Excel to "Text (tab delimited)" format. (Previously, only the CSV (comma delimited) format was acceptable to NM-TRAN.) Carriage return is typically seen as "^M" at end of lines when a file is copied from MS-DOS to Unix with ftp's binary option or as an email attachment.

Now, NMTRAN will permit such characters, as follows:

1) In the NMTRAN control stream, all such characters are converted to spaces, with the exception of verbatim code, which is left as-is.

2) When NMTRAN is reformatting the input data set (i.e., there is no format specification), they are treated as field delimiters, exactly the same as a comma.

E.g., let T stand for the tab character.

"1T2" and "1,2" are read as "1 2"

"1TT2" and "1,,2" are read as "1 0 2"

3) When there is a format specification for the input data set, NMTRAN attempts to detect low-value characters and warn the user. Low-value characters at the end of a line of data are ignored, unless the line is otherwise blank. This too is platform dependent.

Thanks to Steve Shafer for suggesting these changes. They have been tested on Solaris, Linux, and MS-DOS, on ASCII computers. It is expected that they will work the same on EBCDIC computers. However, the user should check carefully what happens on a particular platform before including tabs in files for NMTRAN.

(See the \$DATA Help Item.)

30. \$DATA record (RECORDS option)

A change has been made to option RECORDS=n of the \$DATA record. The value of n may now be as large as 99999999. Previously, n could not be larger than 9999. Note that if the RECORDS=n option is omitted, NM-TRAN reads the file to end-of-file, which with past versions was the only way that a file could have more than 9999 records. Now, this may be specified explicitly.

31. \$PRIOR record (New feature)

A new NM-TRAN record \$PRIOR may be used instead of a user-written PRIOR subroutine.

(See the \$PRIOR Help Item.)

32. \$ESTIMATION record: NOOMEGABOUND, etc.

There are new options THETABOUNDTEST, OMEGABOUNDTEST, SIGMABOUNDTEST, as well as NOTHETABOUNDTEST, NOOMEGABOUNDTEST, NOSIGMABOUNDTEST.

(See the \$ESTIMATION Help Item.)

33. \$ESTIMATION record

The MAXEVALS option may now be as large as 99999999. Previously, the largest value permitted was 9999.

34. \$TABLE and \$SCATTER: G and H elements

Elements of G and H can be displayed in \$TABLE and \$SCATTER without the use of verbatim code. E.g.,

```
$TABLE G11 G21 G31 H11 H21
```

(See the \$TABLE and \$SCATTER Help Items.)

35. Abbreviated code: PHI

The NONMEM utility function PHI may be used in all blocks of abbreviated code. The argument may be a random variable.

(See the PHI and Abbreviated Function Example Help Items.)

36. Abbreviated code: ROCM50

Arrays CNTID and IIDX from ROCM50 may be used on the right in abbreviated code in \$ERROR, \$INFN, \$PK, and \$PRED blocks. When used in a WRITE statement, "IIDX,CNTID" may be specified. Pairs of values are written, one pair per line, one pair for each individual record.

CAUTION:

With NONMEM VI 2.0, do not define common ROCM50 in verbatim code. Verbatim code will no longer access ROCM50 correctly.

(See the ROCM50 example Help Item.)

I.4.2. PREDPP related

1. The variable A_OFLG is a new reserved variable in \$PK abbreviated code, which may be tested as a right-hand quantity (see section 3). When A_OFLG=1, compartment amounts may be initialized:

```
IF (A_OFLG.EQ.1) THEN
  compartment initialization block
ENDIF
```

A_0(i) also is now a reserved variable which may be used as a left-hand quantity in a compartment initialization block. Specifically, A_0(i) is the compartment initialization value of A(i), i.e., the value to which the ith compartment amount is to be set when A_OFLG=1. A_INITIAL(i) is a synonym for A_0(i). The above is referred to as an explicit use of A_OFLG. There may be implicit uses.

(See the COMPARTMENT INITIALIZATION BLOCK Help Item.)

2. Variables A(n) may now be used as right-hand quantities in \$PK abbreviated code, where they denote the latest computed compartment amounts. That is, the A(n) are the amounts at the previous event time, or if at a later time (but before the event time for which it may be that PK is being called) a lagged or additional dose was given, or a regular infusion was terminated, or an event occurred at a modeled event time, then the A(n) are the amounts at the latest such time. The time at which the A(n) are in fact computed is found in the variable TSTATE (from PREDPP read-only common PROCm9), also a right-hand quantity in \$PK.

If ETA variables and A(n) variables are used in \$PK abbreviated code, then any \$OMEGA records referring to these ETA variables should precede the \$PK record, or if an \$MSFI record is used, this record should precede the \$PK record and include the option NPOP=m, where m is the number of population ETA variables.

(See the \$PK and PROCM9 Help Items.)

3. A new NM-TRAN record, \$INFN, allows abbreviated code to substitute for user-written FORTRAN code. NM-TRAN generates the FORTRAN INFN routine (there is also an NM-TRAN Library INFN routine). In an \$INFN block, ICALL may be tested for initialization/finalization values 0, 1 and 3:

```
IF (ICALL.EQ.0) THEN
    run initialization block
ENDIF
```

```
IF (ICALL.EQ.1) THEN
    problem initialization block
ENDIF
```

```
IF (ICALL.EQ.3) THEN
    problem finalization block
ENDIF
```

ICALL may now also be tested for these values in \$PK and \$ERROR blocks, in which case the run initialization, problem initialization, or problem finalization block is moved to an INFN routine. (See \$INFN and INITIALIZATION-FINALIZATION BLOCK Help Items.)

4. \$MODEL and \$PK: I_SS (Initial Steady State feature)

With NONMEM VI 2.0, the \$MODEL record has a new option, I_SS. The \$PK block has a new reserved variable, I_SS.

See I.3.5 above.

(See the PRDPK4 and \$MODEL and \$PK Help Items.)

I.5. REFERENCES

1. Beal, SL. Populations pharmacokinetic data and parameter estimation based on their first two statistical moments. *Drug Metabolism Reviews* 1984; 15: 173-193.
2. Gisleskog, PO, Karlsson, MO, Beal SL. Use of prior information to stabilize a population data analysis. *J. Pharmacokin. Pharmacodyn.* 2002; 29: 473-505.
3. Beal SL. Conditioning on certain random events associated with statistical variability in PK/PD. *J. Pharmacokin. Pharmacodyn.* 2005; 32: 213-243.
4. Wang, Y. Derivation of various NONMEM estimation methods. *J Pharmacokin. Pharmacodyn.* 2007; 34: 575-593.

Appendix I. Control Record Formats

Records marked with * may be continued.

The record name, e.g., "INDX", is not repeated on continuation(s).

FILE record (FILE) (A4,4X,A72)

Field No.	Value	Function
1	NULL	no file stream
	72 chars	name of file stream

SUPER record (SUPR) (A4,4X,I4,I8,I4)

Field No.	Value	Function
1	1-9999	Number of problems in the superproblem
2	2-9999	Number of iterations of the superproblem.
3	0	Input information will be printed for first problem only
	1	Input information will be printed for all problems

PROBLEM record (PROB) (A4,4X,A72)

Field No.	Value	Function
1	72 chars	problem heading

DATA record (DATA) (A4,4X,18I4)

Field No.	Value	Function
1	0-1	Simulation REWIND(1) or NOREWIND(0) from SIM record
2	0 or blank	data set is embedded in the control stream
	1	data set is in a separate file
	-1	re-use the data set from the previous problem.
3	0 or blank	FORTTRAN unit not to be rewound
	1	FORTTRAN unit to be rewound
4	0	data set to be read to FINISH record or end of file
	1-9999	no. of data records (low-order digits)
5	1-PD	no. of data items per data record
6	0	not data checkout
	1	data checkout only
7	0-9999	no. of data records (high-order digits)
		The no. of data records is Field 6 * 10000 + Field 3.
		When Field 6 is 0 or blank, this is simply Field 3

ITEM record (ITEM) (A4,4X,18I4)

Field No.	Value	Function
1	0-PD	index of ID data item
2	1-PD	index of DV data item
3	0-PD	index of MDV data item
4	0-PD	no. of data item indices in INDXS
5	0	no user-supplied labels.
	1	user-supplied labels.
6	0	standard labels PRED,RES and WRES used.
	1	nonstandard labels used.
7	0-PD	index of L2 data item
8	0-PD	index of first data item specified in CONTR record
9	0-PD	index of second data item specified in CONTR record
10	0-PD	index of third data item specified in CONTR record
11	0-50	no. of user-supplied labels for tables, scatters
12	0-PD	index of MRG_ data item
13	0-PD	index of RAW_ data item
14	0-PD	no. of items on OMIT record
15	0-PD	index of RPT_ data item

INDEX record (INDX)* (A4,4X,18I4)

Field No.	Value	Function
1	1-PD	1st element of INDXS
2	1-PD	2nd element of INDXS
	etc.	

XVID record (XVID) (A4,4X,18I4)

Field No.	Value	Function
1	1-PD	XVID1
2	1-PD	XVID2
3	1-PD	XVID3
4	1-PD	XVID4
5	1-PD	XVID5

LABEL record (LABEL)*

The LABEL item contains a comma delimited list of labels, beginning at position 9, 3 items per line, over an unlimited number of lines. The first line contains the item LABEL in column 1, and subsequent lines have blanks in positions 1-4.

LABEL	ID ,	JID ,	TIME
	CONC ,	DOSE ,	RATE
	EVID ,	MDV ,	CMT

The order is as follows:

Field No.	Item
1:	label of 1st data item
2:	label of 2nd data item
m:	label of last data item
m+1:	label for PRED (if ITEM(6)=1)
m+2:	label for RES (if ITEM(6)=1)
m+3:	label for WRES (if ITEM(6)=1)
m+p+1:	label for 1st variable in NMPRD4†
m+p+2:	label for 2nd variable in NMPRD4† , etc.
m+p+q:	label for last displayed variable in NMPRD4

Note

m=no. of data items per data rec.=DATA(4)

p=3 if non-standard labels for PRED, RES, WRES (ITEM(6)=1)

p=0 otherwise

q=no. of user supplied labels for tables, scatters=ITEM(11)

† Blank if this variable is not displayed

STRUCTURE record for LBW1 (LBW1)

The LBW1 item contains a comma delimited list of labels for the additional weighted residual type parameters, starting at position 6 in each line

```
LBW1 IWRS , IPRD , IRS
      NPRED , NRES , NWRES
      NIWRES , NIPRED , NIRES
      CPRED , CRES , CWRES
      CIWRES , CIPRED , CIRES
      PREDI , RESI , WRESI
      IWRESI , IPREDI , IRESI
      CPREDI , CRESI , CWRESI
      CIWRESI , CIPREDI , CIRESI
      EPRED , ERES , EWRES
      EIWRES , EIPRED , EIRES
      NPDE , ECWRES , NPD
      OBJI
```

STRUCTURE record for LTHT (LTHT)

STRUCTURE record for LETA (LETA)

STRUCTURE record for LEPS (LEPS)

Any symbolic labels for THETAS, ETAS, and Epsilons defined in ABBR REPLACE records are placed on the following records, starting in position 9:

```
LTHT      1=THETA ( CL ) , 2=THETA ( V )
LETA      1=ETA ( CL ) , 3=ETA ( V )
LEPS      1=EPS ( Q )
```

OMIT record (OMIT)* (A4,4X,18I4)

Field No.	Value	Function
1	4 chars	no. of 1st data item omitted from template matching
2	4 chars	no. of 2nd data item omitted from template matching
	etc.	

FORMAT record (FORM) (A4,4X,A72/A80)

Field No.	Value	Function
1	80 chars	format specification (field begins on first continuation record)

FIND record (FIND) (A4,4X,18I4)

Field No.	Value	Function
1	0	
2	0	
3	0	No Model specification file (MSFI)
	1	A Model specification file (MSFI) is to be read.
4	0	estimate on file not to be rescaled.
	1	estimate on file to be rescaled.
5	0	No ONLYREAD option
	1	ONLYREAD option
6	0-1	1=NOMSFTTEST

STRUCTURE record for MSFV (MSFV)

Right after the FIND record for MSFI, if it exists, the MSFV (NM74) contains the MSFI file version, starting at position 9:

MSFV 7.2.0

STRUCTURE record for ETA (ETA) (A4,I4,A)

STRUCTURE record for PHI (PHI) (A4,I4,A)

With FILE specified:

Line	Field No.	Value	Function
1	1	0	0 indicates file name is to be given (FILE)
2	1	alpha	FORMAT
3	1	-1-INTBIG	TBLN

With no FILE specified:

Line	Field No.	Value	Function
1	1	1-INTBIG	number of etas or phis listed starting at position 9, on each line, comma-delimited.

initial STRUCTURE record (STRC) (A4,4X,18I4)

Field No.	Value	Function
1	0-999	length of THETA
2	0-999	dimension of OMEGA
3	0-999	dimension of SIGMA
4	blank	
5	blank	
6	0 or blank	OMEGA constrained with a block set partition
	1	OMEGA constrained to be diagonal
7	0 or blank	only if field 6 has value 1
	1-999	number of block sets for OMEGA

If the dimension of SIGMA is 0, the following fields may be ignored.

8	0 or blank	SIGMA constrained with a block set partition
	1	SIGMA constrained to be diagonal
9	0 or blank	SIGMA only if field 8 has value 1
	1-999	number of block sets for SIGMA
10	blank	
11	blank	
12	0 or blank	default THETA boundary test
	1	No default THETA boundary test
13	0 or blank	default OMEGA boundary test
	1	No default OMEGA boundary test
14	0 or blank	default SIGMA boundary test
	1	No default SIGMA boundary test

STRUCTURE record for OMEGA (STRC)* (A4,4X,18I4)

STRUCTURE record for SIGMA (STRC)* (A4,4X,18I4)

Field No.	Value	Function
1	1-999	size of 1st. block set
2	1-999	dimension of blocks in 1st. block set
3	1-999	size of 2nd. block set
4	1-999	dimension of blocks in 2nd. block set
		etc.

THETA CONSTRAINT record (THCN) (A4,4X,18I4)

Field No.	Value	Function
1	0 or blank	THETA unconstrained
	1	THETA constrained
2	0 or blank	use default size of initial. est. search
	1-9999	no. of points to be examined during initial est. search.
3	0 or blank	ABORT if PRED sets error return code to 1 during search
	1	NOABORT - Ignore PRED error return code during search
	2	NOABORTFIRST - Same, even with first values.

THETA record (THTA)* (A4,4X,comma-delimited list)

Field No.	Value	Function
1	initial est. of θ_1	(blank if NONMEM is to obtain the initial est.)
2	initial est. of θ_2	(blank if NONMEM is to obtain the initial est.)
	etc.	

LOWER BOUND record (LOWR)* (A4,4X,(comma-delimited list))

Field No.	Value	Function
1	lower bound for θ_1	
2	lower bound for θ_2	
	etc.	

UPPER BOUND record (UPPR)* (A4,4X,(comma-delimited list))

Field No.	Value	Function
1	upper bound for θ_1	
2	upper bound for θ_2	
	etc.	

DIAGONAL record (DIAG)* (A4,1X,A1,1X,A1,(comma-delimited list))
for OMEGA or SIGMA

Field No.	Value	Function
Pos. 1	0	Diagonals Variance
	1	Diagonals standard deviation (STANDARD)
Pos. 2	blank	Not fixed.
	1	Fixed.
	2	NONMEM is to obtain the initial estimate(s).
1		initial est. of (1,1) element of matrix
2		initial est. of (2,2) element of matrix
	etc.	

BLOCK SET record (BLST)* (A4,1X,A1,1X,,A1,(comma-delimited list))
for OMEGA or SIGMA

Field No.	Value	Function
Pos. 1	0	Diagonals Variance, Off-diagonals covaraince
	1	Diagonals standard deviation, Off-diagonals covaraince (STANDARD)
	2	Diagonals Variance, Off-diagonals correlation (CORRELATION)
	3	Diagonals standard deviation, Off-diagonals correlation (STANDARD CORRELATION)
	4	Cholesky format (CHOLESKY)
Pos. 2	blank	Not fixed.
	1	Fixed.
	2	NONMEM is to obtain the inital estimate(s).
1		initial est. of (1,1) element of matrix
2		initial est. of (1,2) element of matrix
	etc.	
		use symmetric enumeration

SIMLUATION record (SIML) (A4,4X,I2,I3,I2,I13,5I2,I13,I6,I6,1X,A16)

Field No.	Value	Function
1	0 or blank	Simulation Step implemented
	1	Simulation Step not implemented
If the value is 1, the subsequent fields may be ignored.		
2	1-10	no. of random sources (SORC records)
3	0	eta (eps) changes with each record
	1	eta (eps) changes with new ind.rec. (L2 rec) (NEW)
4	0-9999	no. of subproblems
5	0	compute objective function and other steps
	1	only the simulation step
6	0 or blank	no partial derivatives from PRED needed
	1	PRED should compute 1st. derivatives (REQUESTFIRST)
	2	PRED should compute 2nd. derivatives (REQUESTSECOND)
7	0 or blank	simulated observation is Y or F (PREDICTION)
	1	simulated observation is DV (NOPREDICTION)
8	0 or blank	Use inital ests. (TRUE=INITIAL)
	1	with MSFI, use final ests. (TRUE=FINAL)
	2	use values in NMPR16 (TRUE=PRIOR)
9	0-1	1=NOREPLACE
10	-1-INTBIG	BOOTSTRAP
11	0-INTBIG	STRAT data column number
12	0-INTBIG	STRATF data column number
13	alpha	RANMETHOD

ADDITIONAL RECORDS FOR SIML (A4,4X,A)

Record	Function
SFIL	PARAFILE

SOURCE record (SORC) (A4,4X,2A12,I4)

Field No.	Value	Function
1	-1-21474836447	first seed
2	0-21474836447	second seed
3	0 or blank	random numbers are pseudo-normal (NORMAL)
	1	random numbers are pseudo-uniform (UNIFORM)
	2	random numbers are from a nonpar. distrib (NONPARAMETRIC)

SOURCE record (DFLT) (A4,4X,I4)

Field No.	Value	Function
1	-1-1	NOSUB option of DEFAULT record

CHAIN record (CHN)

Line	Format	Field No.	Value	Function
1	(A4,4I12,F12.5,I12)	1	0-2	CTYPE
		2	INTSMALL-INTBIG	SEED
		3	INTSMALL-INTBIG	ISAMPLE
		4	0-INTBIG	NSAMPLE
		5	0.0001-1.0	IACCEPT
		6	0-INTBIG	DF
2	(4X,4I12,A12)	1	INTSMALL-INTBIG	ISAMPEND
		2	0-3	SELECT
		3	0-3	NOTITLE(1,3),NOLABEL(2,3)
		4	-1-INTBIG	DFS
		5	ALPHA	RANMETHOD

ADDITIONAL RECORDS FOR CHAIN (A4,4X,A)

Record	Function
CFIL	FILE
CDLM	FORMAT
ORDR	ORDER
CHFL	PARAFILE

LEVEL record (OLEV)

The OLEV (NM73) contains parameters to the LEVEL command. The data column name pertaining to the level is in columns 9 to 28, and the level description is at position 29-80, for example:

```
OLEV    SID          3[1],4[2]
OLEV    CID          5[3],6[4]
```

If the level description is too long, another line continues the description, but with data column name missing (column 9-28), to indicate it is continuing from the previous line:

```
OLEV    SID          3[1],4[2],5[6],12[4],
OLEV          27[45],67[23]
```

ANNEAL record (ANNL)

The ANNL (NM73) contains parameters to the ANNEAL record, with omega element, followed by its starting value (A4,X4,A6,A6):

```
ANNL    3          4
ANNL    6          0.3
etc.
```

ESTIMATION record (ESTM) (A4,4X,18I4)

Field No.	Value	Function
1	0 or blank	Estimation Step implemented
	1	Estimation Step not implemented

If the value is 1, the subsequent fields may be ignored.

Field No.	Value	Function
2	0-9999	maximum no. of function. evaluations (low-order digits)
	-1	Reuse the value from the previous run (with MSFI)
3	1-8	number of significant figs. required in final est.
4	0 or blank	no summarization of iterations
	n>0	every nth iteration summarized
5	0 or blank	no second search (REPEAT)
	1	second search (REPEAT) implemented
6	0 or blank	MSF not output
	1	MSF output
7	0 or blank	First order (FO) method
	1	Conditional method (METHOD=COND)
8	0 or blank	No POSTHOC etas are to be estimated.
	1	POSTHOC etas are to be estimated.
9	0 or blank	Etas are 0 for comp. of intraind. error (NOINTERACTION)
	1	Nonzero etas for comp. of intraind. error INTERACTION
10	0 or blank	Do not use Laplacian method.
	1	Laplacian method is to be used.
11	0 or blank	ABORT if PRED sets error return code to 1
	1	NOABORT - Attempt theta-recovery when PRED error code 1.
	2	NOHABORT - Attempt recovery even at first iteration
12	0 or blank	Faster method of computation (NOSLOW)
	1	Slower method of computation (SLOW)
	2	Slower method of computation (SLOW=2); for Stieltjes
13	0 or blank	avg. cond. est. of etas unconstrained (NOCENTER)
	1	avg. cond. est. of etas constrained close to 0. (CENTER)
14	0 or blank	First-order model not used (NOFO)
	1	First-order model used with METHOD=1 CENTERING (FO)
15	0 or blank	Second eta-derivs. computed by PRED (NONNUMERICAL)
	1	Second eta-derivs. for Laplacian to be obtained numerically.
16	0 or blank	Y or F (with user-supplied code) is a prediction.
	1	Y or F is a LIKELIHOOD.
	2	Y or F is a -2LOGLIKELIHOOD
17	0 or blank	Not the Hybrid method
	1-99	no. of etas fixed to zero by ZERO recs. (Hybrid method)
18	0 or blank	Not the Stieltjes method.
	1	Stieltjes method; no GRID option.
	2	Stieltjes method; GRID was specified.

ESTIMATION rec. continuation rec. () (A4,4X,18I4)

Field No.	Value	Function
1	0 or blank	Required if estimation step is omitted, otherwise:
	0 or blank	The REPEAT2 option is not coded; same as NOREPEAT2

Introduction to Version VI

	1	REPEAT2 (with Stieltjes)
2	0 or blank	No ETABARCHECK.
	1	ETABARCHECK option is coded.
3	0 or blank.	Sum contrib. to obj. func. in data set order.
	1	Sort contrib. to obj. func. prior to sum (SORT)
4	0-9999	maximum no. of function evaluations (high-order digits) The no. of func. evals. is Field 4 * 10000 + low-order When Field 4 is 0 or blank, this is simply low-order
5	1-15,100	SIGL
6	1-15,100	SIGLO

BAYES ESTIMATION record		(BEST)		
Line	Format	Field No.	Value	Function
1	(A4,4I12,F12.5,I12)	1	-1-16	BAYES METHOD
			<=0	FO/FOCE/Laplace
			10	DIRECT
			11	BAYES
			12	ITS
			13	SAEM
			14	IMP
			15	IMPMAP
			16	CHAIN
		2	-1-INTBIG	PSAMPLE_M1
		3	-1-INTBIG	PSAMPLE_M2
		4	0-INTBIG	PSAMPLE_M3
		5	0.0001-1.0	PACCEPT
		6	-1-INTBIG	OSAMPLE_M1
2	(X4,2I12,F12.5,3I12)	1	-1-INTBIG	OSAMPLE_M2
		2	0-INTBIG	OSAMPLE_M3
		3	0.0001-1.0	OACCEPT
		4	1-INTBIG	ISAMPLE/ICHAINS (non-CHAIN)
			INTSMALL-INTBIG	ISAMPLE (CHAIN)
		5	0-INTBIG	ISAMPLE_M1
		6	0-INTBIG	ISAMPLE_M2
3	(X4,I12,F12.5,4I12)	1	0-INTBIG	ISAMPLE_M3
		2	0.0-1.0	IACCEPT
		3	0-INTBIG	NSAMPLE/NITER
		4	0-INTBIG	NBURN
		5	0-INTBIG	DF
		6	0-1	EONLY
4	(X4,5I12,F12.5)	1	INTSMALL-INTBIG	SEED
		2	0-1	NOPRIOR
		3	0-3	NOTITLE(1,3),NOLABEL(2,3)
		4	0-4	CTYPE
		5	1-INTBIG	CITER/CNSAMP
		6	0.0000001,1	CALPHA
5	(X4,4I12,2E12.5)	1	0-INTBIG	CINTERVAL
		2	0-INTBIG	MAPITER
		3	-1-INTBIG	MAPINTER
		4	0-INTBIG	ISAMPLE_M1A
		5	0-INF	ISCALE_MIN
		6	0-INF	ISCALE_MAX
6	(X4,5I12,A12)	1	0-INTBIG	CONSTRAIN
		2	0-15	ATOL
		3	0-2	FNLETA
		4	0-2	OPTMAP
		5	0-3	ETADER
		6	ALPHA	RANMETHOD
7	(X4,5I12,E12.5)	1	0-INTBIG	MCETA
		2	0-2	NONINFETA
		3	INTSMALL-INTBIG	ISAMPEND (CHAIN)

Introduction to Version VI

		4	0-1	ETASTYPE
		5	0-1	AUTO
		6	0.0-1000	STDOBJ
8	(X4,I12,2E12.5,I12,I12,I6)	1	0-3	NUMDER
		2	0-INF	PSCALE_MIN
		3	0-INF	PSCALE_MAX
		4	-1-INTBIG	DFS
		5	0-3	SELECT
		6	0-1	NOCOV
9	(X4,I6,I6)	1	0-1	DERCONT
		5	-1-1	NOSUB

ADDITIONAL RECORDS FOR ESTIMATION (A4,4X,A)

Record	Function
BFIL	FILE
BDLM	FORMAT/DELIM
BMUM	MUM (may go beyond 80 characters)
BGRD	GRD (may go beyond 80 characters)
ORDR	ORDER
PFIL	PARAFILE
FFIL	FPARAFILE

ZERO record (ZERO)* (A4,4X,18I4)

Field No.	Value	Function
1	0	conditional estimate for eta(1)
	1	eta(1) is fixed to 0 (HYBRID method)
2	0	conditional estimate for eta(2)
	1	eta(2) is fixed to 0 (HYBRID method)
	etc.	

GRID record (GRID) (A4,4X,9A8)

Field No.	Value	Function
1	nr	as specified in GRID=(nr,ns,r0,r1)
2	ns	as specified in GRID=(nr,ns,r0,r1)
3	r0	as specified in GRID=(nr,ns,r0,r1)
4	r1	as specified in GRID=(nr,ns,r0,r1)

NONPARAMETRIC record (NONP) (A4,4X,6I4,I12,I6,I6)

Field No.	Value	Function
1	0 or blank	Nonparametric step implemented conditionally
	1	Nonparametric step implemented unconditionally
2	0 or blank	use nonparametric estimate from input MSF
	1	recompute nonparametric estimate
3	0 or blank	obtain marginal cumulatives
	1	compute conditional nonpar. etas (CNPE ETAS)
4	0 or blank	no model specification file is output
	1	a model specification file is output
5	0,1	1=BOOTSTRAP
6	0-3	EXPAND(1,3),NPSUPPE(2,4)
7	0-INTBIG	NPSUPP(E) value
8	1-INTBIG	STRAT data column number
9	1-INTBIG	STRATF data column number

COVARIANCE record (COVR) (A4,4X,18I4)

Field No.	Value	Function
1	0 or blank	Covariance Step conditionally implemented
	1	Covariance Step unconditionally implemented
	2	Covariance Step not implemented
2	0 or blank	covariance matrix set to (R inverse) S (R inverse)
	1	covariance matrix set to R inverse
	2	covariance matrix set to S inverse
3	0 or blank	neither R nor S printed.
	1	R matrix printed
	2	S matrix printed
4	0 or blank	both R and S printed
	1	eigenvalues not printed
	1	eigenvalues printed.
5	0 or blank	default computation.
	1	Special computation with a recursive PRED subroutine.
6	0 or blank	Print Covariance Step arrays in normal format.
	1	Print Covariance Step arrays in compressed format.
7	1	
8	0 or blank	
9	0 or blank	Normal method of computation
	1	Slower method of computation (SLOW)

Introduction to Version VI

Additional COVARIANCE record (COVT) (A4,4X,18I4)

Field No.	Value	Function
1	1-15,100	SIGL
2	1-15	TOL
3	1-15,100	SIGLO
4	1-15	ATOL
5	0-1	1=NOFCOV
6	0-1	1=COVRESUME

Additional COVARIANCE record (CPAR) (A4,4X,A)

Record	Function
CPAR	PARAFILE

initial TABLE record (TABL) (A4,4X,18I4)

Field No.	Value	Function
1	0 or blank	Table Step conditionally implemented
	1	Table Step unconditionally implemented
	2	Table Step not implemented

If the value is 2, the next field may be ignored, and there should not appear any individual TABLE records.

2	1-10	number of tables
---	------	------------------

Add. initial TABLE record (PPAR) (A4,4X,A)

Record	Function
PPAR	PARAFILE

Introduction to Version VI

individual TABLE record (TABL) (A4,1X,3A1,I4,9(I6,I2))

Field No.	Value	Function
Pos. 1	blank	no option record.
	1	an option record follows. (only if at least one item on the option rec. is non-blank)
1	0-PDT	number of selected data item types
2	1-9999	index of 1st selected data item type
3	0-8	sort code for data items of 1st selected type
	-1	Exclude-by item marked by -1
4	1-9999	index of 2nd selected data item type
5	0-8	sort code for data items of 2nd selected type
	etc.	

individual TABLE rec. contin. rec. ()* (A4,1X,3A1,I4,9(I6,I2))
(as needed)

Field No.	Value	Function
1	1-999	index of 9th. selected data item type
2	0	
3	1-999	index of 10th. selected data item type
4	0	
	etc.	

individual TABLE record option rec. () (A4,4X,5I4,I12,I12,A12,I2,I2,1X,A)

Field No.	Value	Function
1	blank	Every data record appears in the table.
	1	Only the first data rec. from each ind. rec. (FIRSTONLY)
2	1	With TABLE file, no printed table (NOPRINT)
	2	With TABLE file, printed table appears in the NONMEM output.
3	bit 0 set	ONEHEADER
	bit 1 set	NOHEADER
	bit 2 set	NOTITLE
	bit 3 set	NOLABEL
4	blank	The TABLE file is opened and is positioned at the start.
	1	The TABLE file is positioned at the end (FORWARD)
5	blank	DV, PRED, RES, WRES appear automatically
	1	DV, PRED, RES, WRES do not appear unless listed (NOAPPEND)
6	INTSMALL-INTBIG	SEED
7	3-INTBIG	ESAMPLE
8	alpha	RANMETHOD
9	0-1	WRESCHOL
10	-1-1	NOSUB
11	alpha	FORMAT

Introduction to Version VI

Additional TABLE records (A4,4X,A)

Record	Function
FRML	LFORMAT (record may be longer than 80 characters)
FRMR	RFORMAT (record may be longer than 80 characters)

initial SCATTERPLOT record (SCAT) (A4,4X,18I4)

Field No.	Value	Function
1	0 or blank	Scatterplot Step conditionally implemented
	1	Scatterplot Step unconditionally implemented
	2	Scatterplot Step not implemented

If the value is 2, the next field may be ignored, and there should not appear any individual SCATTERPLOT records.

2	1-20	number of families
---	------	--------------------

individual SCATTERPLOT record (SCAT) (A4,4X,9I8/8X,9I8)

Field No.	Value	Function
1	1-23	index of data items plotted on abscissa axis
2	1-23	index of data items plotted on ordinate axis
3	0 or blank	a single scatterplot
	1	a one-way partitioned scatterplot
	2	a two-way partitioned scatterplot

If the value of field 3 is 0 or blank, the next two fields should be ignored.

4	1-23	index of 1st separator
---	------	------------------------

If the value of field 3 is 1, the next field should be ignored.

5	1-23	index of 2nd separator
6	0 or blank	no unit slope line appears
	1	unit slope line appears
7	0-99999999	no. of the first data rec. for the scatter (FROM)
8	0-99999999	no. of the last data rec. for the scatter (TO)
9	0 or blank	a line through zero on the ordinate axis if appropriate.
	1	a line through zero on the ordinate axis. (ORD0)
	-1	no line through zero on the ordinate axis.
10	0 or blank	a line through zero on the abscissa axis if appropriate.
	1	a line through zero on the abscissa axis. (ABS0)
	-1	no line through zero on the abscissa axis.
11	0 or blank	Every data record appears in the scatter.
	1	Only the first data rec. from each ind. rec. (FIRSTONLY)
12	0 or blank	Every data record appears in the scatter
	1	Only data records with MDV=0 (OBSONLY).
12	-1-1	NOSUB