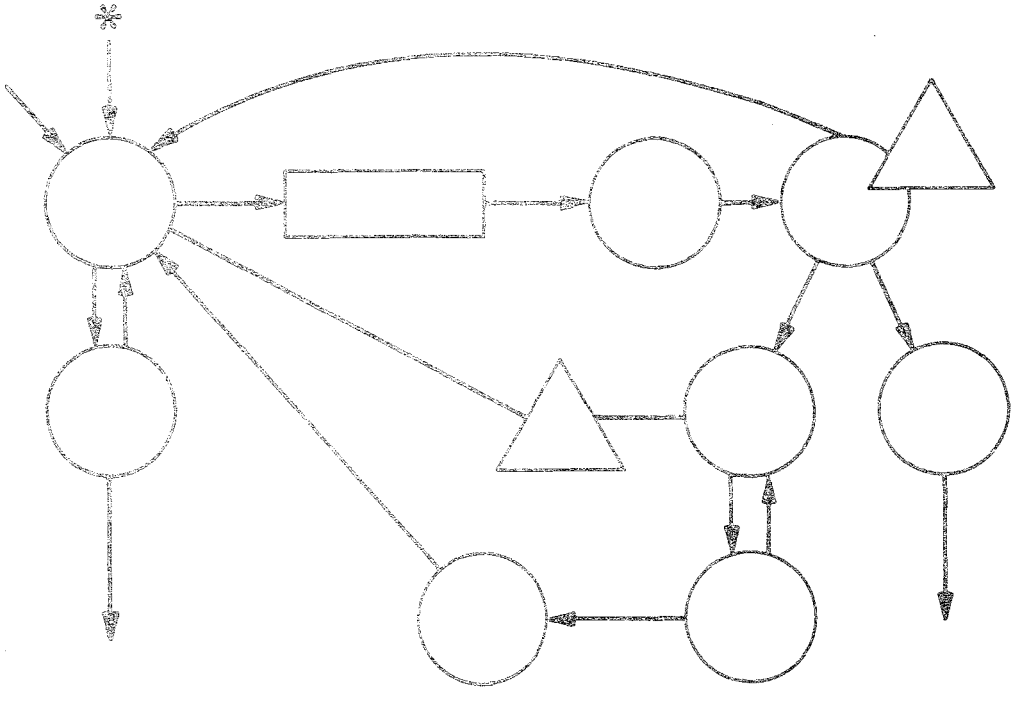
CONSAM

User's Guide



U.S. DEPARTMENT OF
HEALTH AND HUMAN
SERVICES

Public Health Service

National Institutes of Health

CONSAM

User's Guide

By

M. Berman
W. F. Beltz
P. C. Greif
R. Chabay, and
R. C. Boston

Program Version: December 1982
Manual Version: July 1983

Laboratory of Mathematical Biology
National Cancer Institute
National Institutes of Health
Bethesda, Maryland 20205

U.S. DEPARTMENT OF
HEALTH AND HUMAN
SERVICES

Public Health Service

National Institutes of Health

For Administrative Use Only
Revised 21-July-1983

****

TABLE OF CONTENTS

****

INTRODUCTION

CONSAM (short for Conversational SAAM) is an interactive version of the SAAM modeling program (1, 2). CONSAM allows the user, working at an interactive terminal, to create, test, revise and develop mathematical models for simulation and data fitting. Source input files may be used interchangeably by SAAM and CONSAM and problems may be run in either mode.

This manual documents the CONSAM features at the time of publication. CONSAM is being developed on a VAX 11/780 computer but versions are also available for the IBM 1170, DEC-SYSTEM 10, UNIVAC and DEC 20. The full command set described here is available on the most recent VAX version. Conversions to other systems are made periodically. The exact capabilities of your version of CONSAM will depend on the computer system and the date of distribution.

CONSAM results may be displayed in tabular and graphic form. CONSAM currently supports non-graphics terminals, TEKTRONIX 4010 series and 4020 series graphics terminals and the ZETA plotter. The TEKTRONIX 4020 series intelligent terminal is the most sophisticated graphic device currently supported by CONSAM.

This manual consists of 3 parts. Part 1 covers general topics of CONSAM usage. It is suggested that first-time users and users of previous versions read Part 1 and get started using CONSAM. Part 2 contains chapters on specific topics which the user may read or refer to as the need arises. Some of these chapters, such as Chapter 10 on editing, may be needed quite early. These chapters are self-contained in that they do not require knowledge of other Part 2 chapters. Part 3 consists of appendices which are summaries of information contained elsewhere in the manual. These are primarily for quick reference and will be useful to the more advanced user.

Information on running CONSAM on your computer can be obtained from the computer personnel at your installation. Additional information on CONSAM and examples of its use may be found in references 3 and 4.

References:

(1) Berman, M., Shahn, E. and Weiss M.F.: (1962). The routine fitting of kinetic data to models: a mathematical formalism for digital computers. Biophys. J. 2:275-287.

(2) Berman, M. and Weiss, M.F.: (1978). SAAM Manual. U.S. DHEW Publication No. (NIH) 78-180, Washington, D.C.

(3) Boston, R.C., Greif, P.C. and Berman, M.: (1982) CONSAM (conversational version of the SAAM modeling program). In: Lipoprotein Kinetics and Modeling (Berman, M., Grundy, S.M. and Howard, B.V., Eds.). Academic Press, New York.

(4) Boston, R.C., Greif, P.C. and Berman, M.: (1981) Conversational-SAAM - an interactive program for kinetic analysis of biological systems. Computer Programs in Biomedicine. 13:111-119.

(5) Foster, D.M. and Boston, R.C.: (1983) The use of computers in compartmental analysis: the SAAM and CONSAM programs. In: Compartmental Distribution of Radiotracers (Robertson, J., ed.). Chemical Rubber Company, Cleveland.

The authors wish to acknowledge the helpful suggestions of Drs. Loren Zech, David Covell and Meryl Wastney. The excellent technical assistance of Mrs. Sue Smith in compiling this manual is also appreciated.

Part 1

USING CONSAM

****

FUNDAMENTALS

****

This chapter describes a few concepts of CONSAM and of this manual which are essential for effective use of the CONSAM program.

* * * * * COMMAND FORMATS * * * * *

Although the purpose of this manual is to describe each of the commands in CONSAM, a few words need to be said about the commands in general and the conventions used in this manual to describe the format of each command. CONSAM only recognizes upper case characters. Commands names are at most four characters long.

In the command descriptions in this manual:

UPPER CASE characters signify characters which are part of a command, argument or qualifier. These characters must be typed for proper interpretation of the command.

lower case characters signify an argument or qualifier which the user must specify. This may be a number or character string depending on the command. The value entered depends upon the intent of the user.

brackets: [and] enclose an optional argument or qualifier.

braces: { and } enclose a list from which the user must choose one element. Elements within the list are separated with a comma (,).

the following characters: ( ) , = < > # * : / ^ - + and space are used in CONSAM commands. Like UPPER CASE characters, any of theses characters in a command description signifies that that character is to be typed when using the command. There are three exceptions to this rule: a comma (,) within braces ({ and }) is used to separate list elements, a hyphen (-) within braces is used to designate a range of integers, all of which are to be considered list elements, and a hyphen (-) between two lower case characters is used in the name of a user-specified argument or qualifier.

Example:

The command description:

MODE[={1-5,7}]

states that the following commands are valid:

MODE	since the qualifier =value is optional
MODE=7	since 7 is an entry in the list
MODE=2	since 2 is in the range specified by 1-5

and the following commands are not valid:

mode	CONSAM only recognizes upper case letters
MOD	the command name has been abbreviated
MODE=	selection from a list in braces is not optional. An optional list would be written as [{list}].
MODE=6	6 is not an entry in the list

Please note that the actual MODEl command has a longer list of arguments than used in this example.

* * * * * SAAM NOTATION * * * * *

The user describes the model to be used and the data to be fit in a source input file using SAAM notation. A complete description of this "language" may be found in reference (1). For the user to effectively use the finer features of SAAM and CONSAM, it is essential to have this reference. It will be referred to in this guide as the "SAAM Manual".

* * * * * DATA * * * * *

QO refers to observed data values, i.e. the measurements which are obtained in the laboratory and to which a model is to be fit. QC refers to calculated data values, i.e., the model predictions which are the fits to the

observed data. Each datum is assigned to a unique data "component" and a unique T-interrupt (TC) block. These concepts are described in the SAAM Manual. Several CONSAM commands reference data by component or TC block.

* * * * * SWITCHES * * * * *

Results of several CONSAM commands (PLOT in particular) are subject to some options which may be user specified. These options may be SWITched ON or OFF by using the SWITch command. All SWITches have default settings which are reset on each entry to CONSAM. Specific SWITches and the effects of their settings are decribed under the appropriate topics. Appendix B contains a summary of all CONSAM switches, their default settings and their uses. The syntax for the SWITch command is:

SWIT [switch {[ON],O[FF]}]

where:

switch	is the name of a specific switch. If no SWITch is specified, CONSAM will display the current settings of all SWITches.
[ON]	specifies that the switch is to be switched ON. Note that this is the default if no setting is specified.
O[FF]	specifies that the switch is to be switched OFF.

Examples:

SWIT	Display all current SWITch settings.
SWIT LOG ON	Set the LOG switch to the ON setting. This is the same as SWIT LOG and is also the default.
SWIT LOG OFF	Set the LOG switch to the OFF position. This is the same as SWIT LOG O.

The STATus command also displays the current settings of all SWITches. It has the same effect as the SWITch command with no arguments. The syntax for the STATus command is:

STAT Display settings for all switches.

* * * * * GENERAL SOLUTION SEQUENCE * * * * *

Several CONSAM commands require that other commands precede them. This normal solution sequence is shown schematically in Figure 1. It generally follows the solution sequence described in the SAAM Manual except that the user has the ability to step through the computations and restart the sequence at any level which has already been executed. For example, following a PARTials command, the user may "SET" the value of a parameter, then SOLVe, etc.

Figure 1 Legend. (Figure 1 is on facing page)

Solution sequence for basic CONSAM commands. All commands are not shown nor is all available information indicated. The "(GET)" command refers to inspecting the value of a parameter by entering the name of the parameter. The "(SET)" command refers to setting the value of a parameter by entering "parameter=value". See Chapter 3 for details. In general, any result available following a command is also available following any subsequent commands.

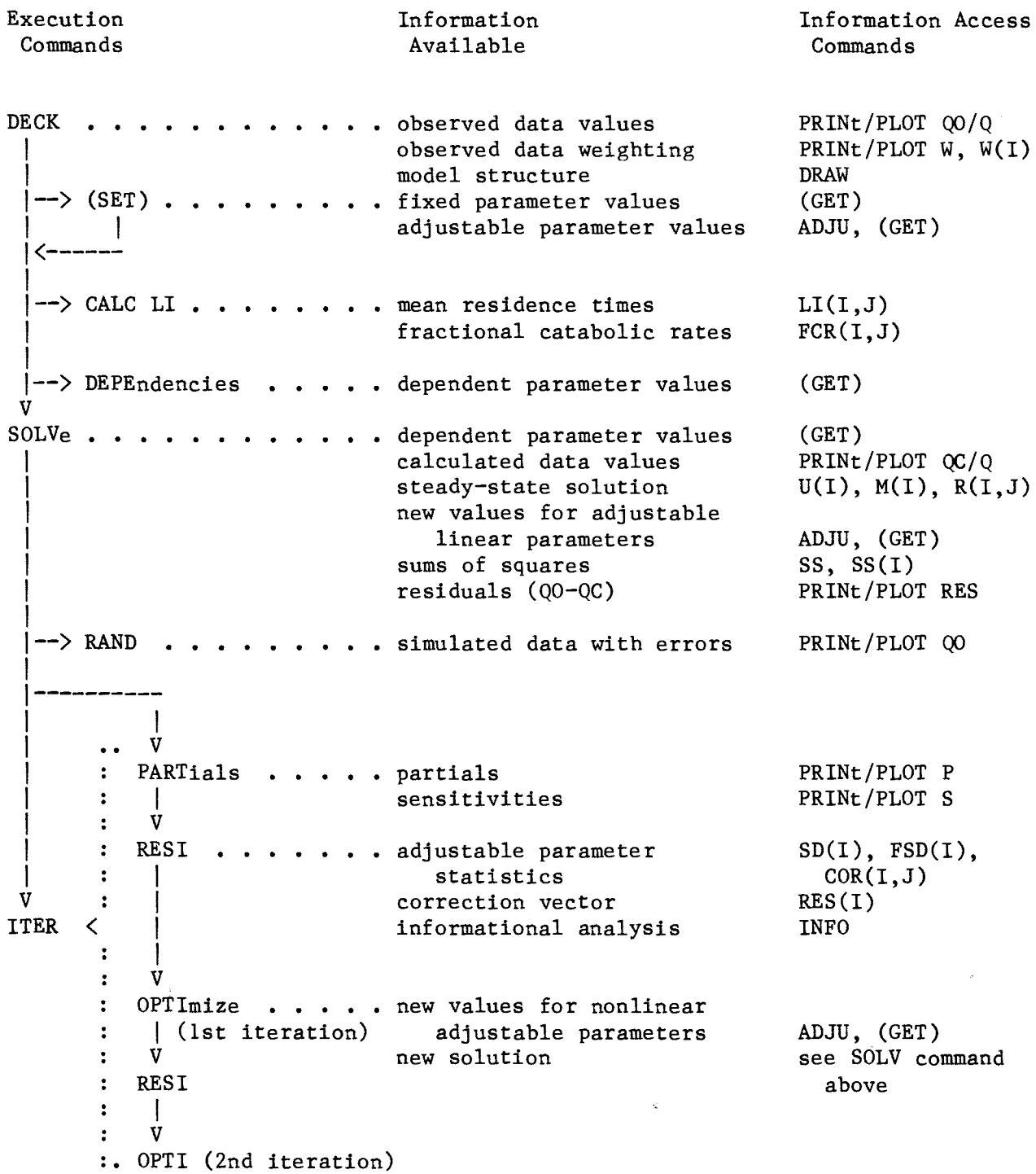


Figure 1. Legend on facing page.

****

GETTING STARTED

****

* * * * * MODEL DESCRIPTIONS * * * * *

Descriptions of the model and data you are working with may exist in two forms:

1. The source input file (sometimes referred to as the DECK or SAAM DECK) contains a description of your problem in SAAM notation. This file can be created before running CONSAM by using one of the system editors or within CONSAM by using the CONSAM editor (see Chapter 10). See reference 1 for information on setting up your problem in SAAM notation.

2. KOMN (common) is the working version of your problem. KOMN is created by CONSAM from a source input file in response to the DECK command and is stored in computer memory (CORE). The information in KOMN is accessed directly using CONSAM commands. KOMN may be permanently saved using the WRITe command (see Chapter 11).

If you have a data file which you wish to use as the source input file, you must make sure that it will be associated with the logical unit 3 before running CONSAM. The exact procedure for doing this is machine dependent. The specific command used to run CONSAM also varies with the computer and system used. For information on running CONSAM, talk to the person who gave you this manual.

* * * * * THE DECK COMMAND * * * * *

Once in CONSAM you must generate KOMN to work with your problem. This is done using the DECK command. The DECK command causes CONSAM to read and process a source input file from logical unit 3. DECK may be used at any time to read and process the current source input file. It is not necessary to use DECK unless the problem structure has been changed. Changes in parameter values can be made in KOMN. The DECK command results in the overwriting of any information in KOMN. If any editing has been performed, the edit buffer is automatically copied into the source input file before processing. The function of the DECK command is shown schematically in Figure 2. The syntax for the DECK command is:

DECK — Process the source input file from logical unit 3 to generate KOMN.

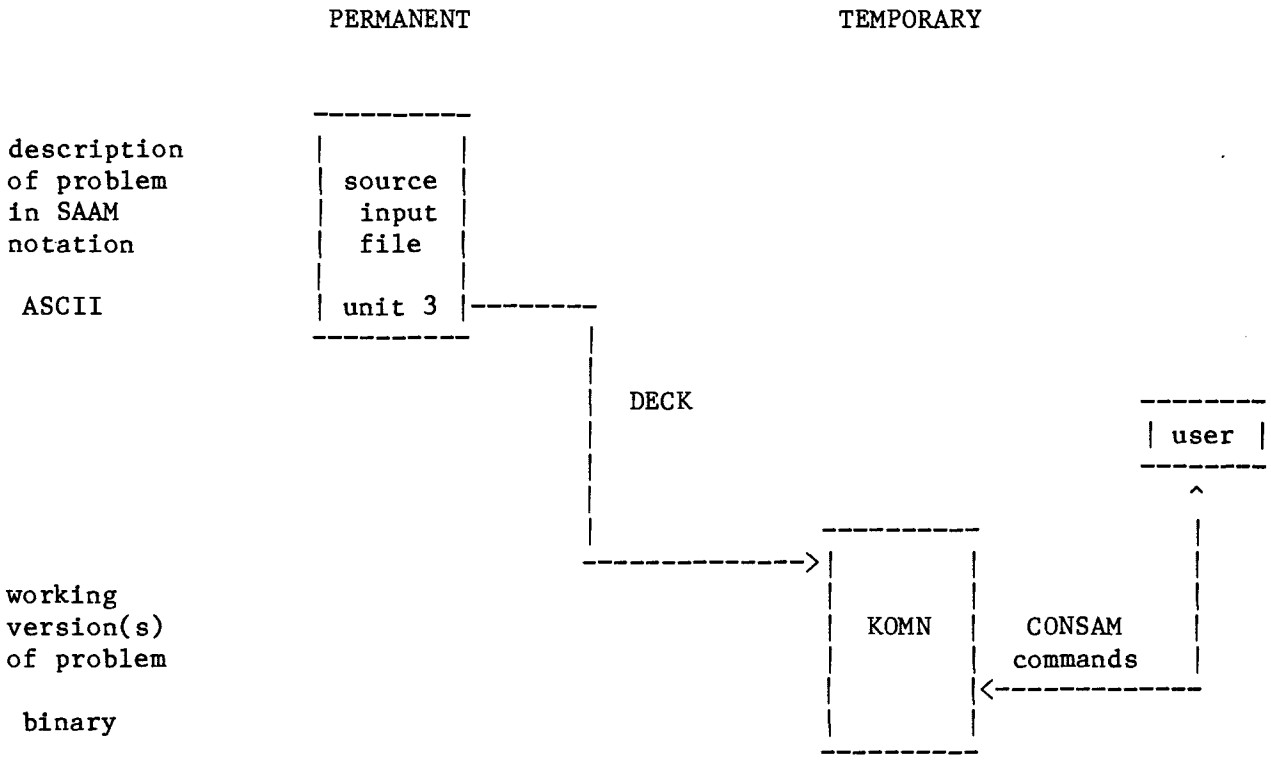


Figure 2.

Schematic diagram showing the relationship between the source input file, KOMN and the user. Note that (most) CONSAM commands affect KOMN, the model version which CONSAM can interpret. To generate KOMN, the DECK command is used to process the source input file.

The TITL and KOMN commands display information about KOMN. These commands may be useful after a DECK to verify aspects of your model:

TITL	Display title of problem read from the A SAAM27 card during source input file processing.
KOMN	Display information on KOMN: creation date and operation used to generate it.
KOMN D	Display detailed information on KOMN: number of parameters, number of data points, etc.

The Q(I), M and M2 "commands" display available data components, the number of data and the number of data plus the number of statistical constraints, respectively:

Q(I)	List the components in KOMN which contain data.
M	Display the number of data points in KOMN.
M2	Display the number of data points plus the number of statistical constraints in KOMN.

****

ABOUT MODEL PARAMETERS

****

There are several types of model parameters. See the SAAM manual for details and the interpretation given to each parameter type by a given model code. Parameters can be of one of three natures: fixed, adjustable or dependent. Fixed parameters are assigned a value and are not altered by CONSAM during solving or data fitting. Adjustable parameters may be adjusted by CONSAM to obtain a better fit to the observed data (as measured by a reduced sum of squares). Dependent parameters are defined in the source input file as explicit or implicit functions of other parameters. Dependent parameters which are functions of an adjustable parameter will in general change values during the data fitting procedure.

Adjustable parameters may be linear (S or K) or nonlinear (all other adjustable parameters). CONSAM adjusts linear parameters during a SOLVe and both linear and nonlinear parameters during an ITERation. Adjustable parameters of both types may be excluded from adjustment by using the EXCLude command. The INCLude command will return the parameter to adjustable status.

* * * * * INSPECTING PARAMETERS * * * * *

The current value for any parameter may be obtained by simply typing the name of the parameter. This will be referred to as the (GET) command. The parentheses indicate that "GET" is not actually typed in when the command is used. The value displayed does not include any function dependencies for fixed or adjustable parameters. Values for dependent parameters may be spurious if a SOLVe or DEPE has not been executed.

Examples:

L(1,2)	Display current value of L(1,2).
L(2,2)	Display the sum of all L(I,2) (including I=0).
S(I,J)	Display current values of all S(I,J).
IC(3)	Display the initial condition for compartment 3.
DT(10)	Display the delay time for compartment 10.

* * * * * CHANGING PARAMETER VALUES * * * * *

A new value for any fixed or adjustable parameter may be specified by entering

parameter=value

This will be referred to as the (SET) command. The parentheses indicate that "SET" is not really typed when using the command. The (SET) command only changes the parameter value in KOMN. It does not affect the edit buffer or source input file.

Examples:

L(2,1)=.1	Set the value of L(2,1) to 0.1
K(2)=1.E-7	Set K(2) to 0.0000001
L(4,4)=3	Multiply all L(I,4) by a constant such that L(4,4)=3.
P(33)=0	Set the value of P(33) to 0.

* * * * * CHANGING THE NATURE OF A PARAMETER * * * * *

The nature (dependent, adjustable or fixed) of a parameter may only be changed by editing and reprocessing (DECKing) the source input file. However, the EXCLude command will cause an adjustable parameter to be treated as if it were fixed. Whether an adjustable parameter is INCLuded or EXCLuded from adjustment is called the status of the parameter. See Chapter 10 for information on editing and Chapter 6 for information on the EXCLude command.

****
SIMULATION (SOLVING)
****

* * * * * SOLVING A MODEL * * * * *

Once you have an error free version of KOMN (see last chapter), you may use the SOLVe command to calculate a solution for the model:

SOLVe — Compute a solution for the model in KOMN.

Execution of the SOLVe command results in the performance of several functions by CONSAM:

1. Dependency equations are solved to generate values for all dependent parameters.
2. The differential equations defined by the model are numerically integrated to generate values for F(I,T) as needed.
3. The steady-state equations are solved to generate values for unknown masses (M's) and inputs (U's).
4. Simulated (QC) values are calculated for all data requested in the source input file.
5. New values are calculated for all included, adjustable linear parameters to obtain a least squares fit of the model simulation to the weighted data.
6. Residuals are calculated for all data as the differences between observed values and calculated values.
7. The weighted sum of squares is calculated for each data component containing weighted data.

* * * * * SOLVING PARAMETER DEPENDENCIES * * * * *

It is possible to solve the parameter dependency equations without taking the time to integrate the differential equations or perform all the other functions associated with the SOLVe command:

DEPE — Solve for values of dependent parameters. Note that SOLV also performs the function of the DEPE command. It is not necessary to use DEPE before SOLV.

* * * * * SOLUTION PROCEDURES AND MODEL CODES * * * * *

Several model solution procedures or model types are available for calculating solutions to the model described in the source input file. For each model type there is a unique number called the model code. A particular model type may be requested by specifying the appropriate model code in columns 49-50 on card 2 of the source input file. If no model code is specified, CONSAM will select a default model type during processing of the DECK command. The model type selected will depend on the information in the source input file. The default model type may be overridden with the MODEl command. Each DECK will reset the model type. All model types are not appropriate for all problems. See the SAAM Manual for details.

The syntax for the MODEl command is :

MODE[={1-5,7,8,10,11,21-23}]

where:

the number from the list is the model code desired. If no model code is specified, then CONSAM will display the current model code. The following model codes are available in CONSAM:

model code	solution procedure
1	4-th order Runge-Kutta
2	Analytic Compartment Solver
3	Similarity Transformation
4	Algebraic Equation Solver
5	Linear Combination of Spectra
7	Chemical Equilibrium Solver
8	Nonlinear Differential Equation Solver (modified Berman-Chu Method)
10	Berman-Chu Convolution Method
11	Population Variance Analyzer
21	Runge-Kutta 4/5 Method
22	Adams Predictor Corrector
23	Gear Stiff Solver

Examples:

MODE	Display the current model code.
MODE=23	Set solution procedure to Gear Stiff Solver.

The LIST command with the MODEl argument may be used to display the available model codes:

LIST MODE	List the MODEl codes available.

****

DISPLAYING RESULTS

****

* * * * * PLOTTING AND PRINTING DATA COMPONENTS * * * * *

Calculated data (QC's), observed data (QO's), residuals (RES's), ratios of calculated to observed data (R's) and statistical weights assigned to data may be displayed using the PRINt and PLOT commands. The PLOT and PRINt commands are very powerful and have many options and display characteristics available to the CONSAM user. These featues are discussed in detail in Chapter 12. It is not necessary to understand all the options available to PLOT data. All scaling options have default values which allow simplified PLOTting for the novice user. The simplest syntax for the PLOT command is:

PLOT data-type(component[,...])

where:

data-type — is the type of data to be PLOTted. Available data-types are:

- QO — observed data values. Observed data are plotted on graphics terminals as open symbols.
- QC — calculated data values. Calculated data are plotted on graphics treminals as continuous lines.
- Q — both observed and calculated data values.
- RES — the residual QO-QC.
- FSE — fractional standard error.
- R — the ratio QC/QO.
- W — statistical weights assigned to the data. The QO's are plotted as open symbols and the W's as a continuous line.
- TH — Theta: second independent variable.

component — is the number of a data component to be plotted. Up to 5 components may be listed in one PLOT command. All data will be plotted on the same graph.

The syntax for the PRINT command is identical to that of the PLOT command:

PRIN data-type(component[,...])

* * * * * SUMS OF SQUARES * * * * *

Sums of squares may be displayed by using the (GET) command:

SS	Display the total weighted sum of squares.
SS(I)	Display the sum of squares for each data component with a non-zero value.
SS(component)	Display the sum of squares for the specified component.

where:

component is the number of a data component, i.e., an integer between 0 and 25.

Example:

SS(3) Display the sum of squares for component 3.

* * * * * STEADY-STATE SOLUTION * * * * *

Steady-state solution results for TC block 0 may be obtained by using the (GET) command:

M(I)	Display the steady-state masses for all compartments.
M(compartment)	Display the steady-state mass for the specified compartment.
U(I)	Display steady-state inputs for all compartments.
U(compartment)	Display the steady-state input for the specified compartment.
R(I,J)	Display all steady-state mass transport rates R(I,J)=M(J)*L(I,J).

R(I,compartment) Display all steady-state mass transport rates out of the specified compartment.

R({compartment,0},J)

Display all steady-state mass transport rates into the specified compartment (or out of the system if the first argument is 0).

R(compartment,compartment)

Display the total steady-state mass transport rate out of the specified compartment.

R({compartment-1,0},compartment-2)

Display the steady-state mass transport rate from compartment-2 into compartment-1 or out of the system.

where:

compartment is the number of a compartment (an integer between 1 and 25).

Examples:

M(5)	Display the steady-state mass for compartment 5.
U(1)	Display the steady-state input for compartment 1.
R(I,3)	Display all steady-state mass transport rates out of compartment 3.
R(7,7)	Display the total steady-state mass transport rate out of compartment 7.
R(4,2)	Display the steady-state mass transport rate from compartment 2 into compartment 4, M(2)*L(4,2).

* * * * * FUNCTION ACCESS * * * * *

It is possible to display stored (QC) values of functions such as UF(I,T), F(I,T), and G(I,T) for the 'current' time, T, and the 'current' TC (time-interrupt) block. Values for explicit analytic functions which have not been stored may be evaluated and displayed. A generic time indicator 'T' may be used to display all calculated values of a function for the current TC block.

The syntax for function access version of the (GET) command is:

function(index[,{time,T}][,C:component])

where:

function	is one of the generic function types F, FF, G, QQ or UF. The function type is required.
index	is an integer designating which specific function of the designated type is to be accessed. The function index is required.
time	is a particular time (independent variable) for which the function value is desired.
T	is a wild card requesting that function data at all available times be displayed.
component	is a data component which is to be searched for function values.

In response to the function access command, CONSAM will first search the data vector for a datum matching the requested function, time and component. This search occurs in order of increasing time through the current TC block. If no TC block is set (see below), all TC blocks will be searched in numerical order. If no time value is specified, CONSAM will accept the first datum matching the requested function and component. If no component is specified, CONSAM will accept the first datum matching the requested function and time. If no component or time is specified, CONSAM will accept the first occurence of the requested function regardless of the time value or component for that datum. If the wild card time value 'T' is specified, CONSAM will display any matches with the specified function and component and continue the search for more matches. If the search fails (i.e., CONSAM does not find at least one datum in the current TC block for the specified function at the specified time), CONSAM will evaluate the function if it is explicitly defined and display the result.

If a parameter value is changed using the (SET) command, the new value will not be used in function calculations unless a DEPEndencies or SOLVe has been performed. Note also that function values found in data components may be scaled by a K parameter while those calculated by CONSAM will not be scaled.

Examples:

F(2)	Display value of F(2) and its error for current T and TC.
G(3,3)	Display value of function G(3) at T=3. (T is not reset).
FF(16,T)	Display all values FF(16) evaluated during last solve. (CONSAM will not evaluate functions when the 'T' argument is used)
F(7,T,C:3)	Display all values of F(7) assembled in component 3.
UF(6,42)	Display UF(6) at T=42. If no such value was calculated at last solve and UF is explicit, then calculate and display a value for UF(6) at T=42.
QO(7,7)	Display the value of the function QO(7) at T=7. Evaluate the explicit function if necessary.

The current time (T) and the current time-interrupt (TC) block may be displayed or specified using the (GET) and (SET) commands respectively:

T	Display current time for function access.
T=time	Set time for function access.
TC	Display current TC block for function access.
TC=time	Set TC block for function access.

****

MODELING AND DATA FITTING

****

Data fitting involves changing the values of some or all model parameters to give a better fit to the data (the QC's become closer to the QO's). There are three methods by which this is usually done using CONSAM.

* * * * * USER DEFINED VALUES * * * * *

The easiest way to alter parameters is to use the (SET) command (page 3.2). Fitting the observed data (QO's) by manually setting a parameter value requires some knowledge about the effect the given parameter has on the solution (QC's). This method s however a good way to develop a "feel" for the model and may be necessary to get the solution close enough to the data for CONSAM to ITERate (see below). This method also has the advantage that the values of fixed parameters may be changed to test the effects of these parameters on the solution.

* * * * * PSEUDO-PARTIALS * * * * *

The second method of data fitting is the use of pseudo-partials. This method takes advantage of the observed improvement in fit (or lack of) between two SOLVes and predicts new estimates for all included adjustable parameters which were manually altered between the two solves. Pseudo-partials projections will only be made for those SOLVe's for which only included adjustable parameters have been altered since the preceding SOLVe. If the value of any excluded or fixed parameter has been changed since the last SOLVe, no pseudo-partials will be calculated. To use the pseudo-partials feature the PART switch must be ON:

SWIT PART ON	During each subsequent SOLVe: calculate pseudo-partials, estimate new values for any included adjustable parameters which were changed since the preceding SOLVe and display these estimates.

Pseudo-partials calculation will not automatically result in new adjustable parameter values. You must use PROJect to cause replacement of the current values with the projected values:

PROJ	Set parameter values to those estimated and displayed during execution of the previous SOLVe.

* * * * * ITERATING * * * * *

The third method of parameter adjustment is the use of the ITERate command. This command invokes the nonlinear least-squares machinery described in the SAAM Manual. Usage of the ITERate command requires that the current solution be close enough to a fit of the data so that changes in the parameters will make a significant improvement. Iterations are done in pairs, partials are calculated only for the first iteration of each pair. The syntax for the ITERate command is:

ITER — Perform 2 complete iterations, allowing all included, adjustable parameters to change.

A single iteration may be divided into 3 steps, which must be executed in the following sequence:

PART — Compute partial derivatives of all components with respect to all included adjustable parameters.

RESI — Calculate the correction vector RES(I) based on current partials matrix.

OPTI — Apply the current correction vector RES(I) and adjust its magnitude to better fit the data.

* * * * * LISTING ADJUSTABLE PARAMETERS * * * * *

The ADJUstable command displays information about adjustable parameters. The display includes parameter name, current parameter value and the upper and lower parameter limits. The letter 'U' or 'B' is appended when a parameter is at its upper or lower limit, respectively. The character '@' is appended for any parameter which is currently EXCLuded from adjustment. The syntax for the ADJUstable command is:

ADJU [{DT,K,L,P,S,UF}]

where:

DT, K, L, P, S or UF

specifies the generic type of the adjustable parameters to be displayed. If the argument is omitted, CONSAM will display all adjustable parameters.

Examples:

ADJU	List all adjustable parameters with their current, minimum and maximum values. Indicate parameters which are at their upper or lower limits and parameters which are EXCLuded.
ADJU L	List current, minimum and maximum values, etc. for all adjustable L parameters.

* * * * * ADJUSTABLE PARAMETER STATISTICS * * * * *

After an iteration or PARTial, RESIdual sequence, statistics on adjustable parameters are available. This information may be retrieved by using the (GET) command. During processing of the source input file each adjustable parameter is assigned an index between 1 and the number of adjustable parameters. Statistical information for a specific parameter may be requested by specifying the index of that parameter and information for all adjustable parameters may be requested by using the wild card indices I and J. Adjustable parameter statistics may not be available following the ITER command if the RES vector for the second iteration is 0 (i.e., you are too close to a local minimum). To generate a nonzero RES vector, use the sequence SOLV, PART, RESI. Statistics are not available for EXCLuded parameters. The following information is available for included adjustable parameters:

SD({index,I})	Standard deviation of adjustable parameter estimate(s).
FSD({index,I})	Fractional standard deviation of adjustable parameter estimate(s).
COR({index,I},{index,J})	Correlation between adjustable parameter estimates.
RES({index,I})	RES vector component(s).

where:

index	is the index of the adjustable parameter for which the appropriate information is to be displayed.

* * * * * USER INFLUENCE ON DATA FITTING * * * * *

The user may affect the outcome of the ITERate command and the pseudo-partials projection by specifying which data are to be weighted and which parameters are to be adjusted.

The UNWEight command requests that specified data be assigned zero weights for subsequent partials, pseudo-partials and sums of squares calculations. The UNWEight command affects KOMN only. Original weights may be restored using the REWEight command and will also be restored in response to the WRITe command. Each datum with a weight of zero is indicated on 4025 plots by a little tiny x in the middle of the OO symbol. The REWEight command restores original weights as determined from the source input file to the specified data. The syntaxes for the UNWEight and REWEight commands are identical:

UNWE name [time[time]]

REWE name [time[time]]

where:

name — is a component number, the name of a data segment or the special name ALL. See Chapter 13 on names and naming. If the name ALL is used, CONSAM will UNWEight or REWEight all data and any time arguments will be ignored.

time — is a value for the independent variable T. If no time arguments are given, CONSAM will UNWEight or REWEight all data in the specified component or name. If one time argument is given, CONSAM will UNWEight or REWEight all data in the specified component or name with T values equal to the time argument entered. If two time arguments are given, CONSAM will UNWEight or REWEight all data in the specified component or name with T values within the interval delimited by the two T arguments (inclusive).

Examples:

UNWE ABQ	Assign weights of 0 to all data in name ABQ.
UNWE 3 1.	Unweight datum at time 1 in component 3.
REWE B 2 10	Reweight data in name B from time 2 to time 10, inclusive.
REWE ALL	Reweight all data.

The EXCLude command requests that one or more adjustable parameters not be included in subsequent partials and pseudo-partials calculations and not be changed during subsequent ITERations or SOLVes. EXCLude applies only to parameters specified as adjustable in the source input file.

The INCLude command requests that one or more adjustable parameters be included in subsequent partials and pseudo-partials calculations and allowed to change during subsequent ITERations and SOLVes. The INCLude command counteracts the EXCLude command and applies only to parameters which were specified as adjustable in the source input file.

The syntaxes for the EXCLude and INCLude commands are identical:

EXCL {parameter[,...],ALL}

INCL {parameter[,...],ALL}

where:

parameter	is a specific adjustable parameter to be EXCLuded or INCLuded. Several parameters may be listed. Generic names (e.g., "L") or wild cards (e.g., "L(I,J)") are not allowed. Each parameter to be EXCLuded or INCLuded must be individually listed unless ALL is specified.
ALL	requests that all adjustable parameters be EXCLuded or INCLuded.

Examples:

EXCL P(5)	Do not include adjustable parameter P(5) in the calculation of partials or allow it to adjust during subsequent ITERations.
EXCL S(3,1),S(3,2)	Do not include S(3,1) and S(3,2) in partials calculations or allow them to adjust during subsequent SOLVes and ITERations.
INCL L(0,1)	Include L(0,1) in partials and allow it to adjust during subsequent ITERations.

The MINimum and MAXimum commands change the lower and upper limits of an adustable parameter respectively. These commands only apply to adjustable parameters and only affect KOMN although new limits will be written to the edit buffer in response to an UPDAte command (Page 10.3). The syntaxes for the MINimum and MAXimum commands are identical:

MIN parameter value

MAX parameter value

where:

parameter — is the name of the parameter to be given a new limit.

value — is a number specifying the new limit to be used.

Examples:

MIN P(1) .001	Set lower limit of P(1) to .001.
MAX L(3,2) 30.	Set upper limit of L(3,2) to 30.

****
ANALYSIS
****

This chapter covers several features in CONSAM which allow the user to inspect KOMN for information about the model, its solution and the observed data.

* * * * * PARTIALS AND SENSITIVITIES * * * * *

The partials matrix a is defined as the matrix in which element a(I,J) is the partial derivitive of the simulated value of the Ith datum (QC(I)) with respect to the Jth adjustable parameter (x(J)): a(I,J) = (dQC(I)/dx(J)). An estimate of the partials matrix may be calculated by using the PARTials command:

PART — Estimate the partial derivative of each calculated datum with respect to each included adjustable parameter. Partials are estimated by finite difference.

The sensitivity matrix s is defined as the matrix in which element s(I,J) is the fractional change in the Ith calculated datum QC(I) given a fractional change in the Jth adjustable parameter x(J):

$$\begin{aligned} s(I,J) &= (dQC(I)/QC(I)) \,/\, (dx(J)/x(J)) \\ &= (dQC(I)/dx(J)) * (x(J)/QC(I)) \\ &= a(I,J) * (x(J)/QC(I)) \end{aligned}$$

Sensitivities for adjustable parameters are therefore calculated from the partials and are available in CONSAM after the PART command has been executed. A comparison of sensitivities at a given solution time gives a good idea of the relative influence of parameters on a solution at that time. Partials and sensitivities may be displayed using the PRINt and PLOT commands. A full description of these commands may be found in Chapter 12. The simple syntax for this form of the PLOT command is:

PLOT {P,S}(component[,...]) parameter[,...]

where:

P — specifies that partials are to be PLOTted.

S — specifies that sensitivities are to be PLOTted.

component — is a data component for which partials or sensitivities are desired. Up to five components may be listed in one PLOT command.

parameter — is the name of a specific adjustable parameter for which partials or sensitivities are desired. Up to 5 parameters may be listed.

The syntax for PRINting partials or sensitivities is identical to that for the PLOT command:

```
PRIN {P,S}(component[,...]) parameter[,...]
```

* * * * * DATA SIMULATION * * * * *

The RAND command requests CONSAM to calculate new observed data values (QO's) for a specified data component. These simulated QO values are calculated as the current QC values with randomly generated errors added in. Since the QC values are generated from the model in KOMN, this gives the user the capability to analyze data generated from a known model. The user has the option of specifying the seed for the random number generator and the error structure of the calculated errors. The syntax for the RAND command is:

```
RAND[/distribution][/{FSD,SD,RQO}[=value]][/SEED=seed][/BIAS=bias] Q(component)
```

where:

distribution — specifies the error distribution to be used. The following error distributions are available:

NORM	normal (Gaussian)
EXP	exponential
RECT	rectangular (uniform)

If no distribution is specified, the default is NORM.

SD — is the standard deviation of the error distribution, the default SD is 1.

FSD — is the fractional standard deviation of the error distribution, SD=FSD*QC, the default FSD is 0.1

RQO — specifies that SD=RQO*SQRT(QC), the default RQO is 1.

If SD, FSD or RQO is not specified, the default standard deviation is dependent upon the distribution. For a NORMal distribution, the default is FSD=0.1. For an EXPonential or RECTangular distribution, the default is SD=1.

value — is a number to be used for SD, FSD or RQO instead of the default value.

seed — is a seed value for the random number generator. The default value will be different with each execution of the RAND command. The user may specify a seed to be used to assure the regeneration of the same errors in more than one RAND execution. The seed must be a number between 0 and 1.

bias — is the mean of the error distribution. If no bias is specified a default mean of 0 will be used.

component — is the number of a data component for which simulated data are to be generated.

Examples:

RAND/SEED=0.121 Q(3)	Generate QO(3) values with the following structure: QO(3)=QC(3) + 0.1QC(3)x where x is N(0,1).
RAND/EXP/SD=.5 Q(17)	Generate QO(17) values with the following structure: QO(17)=QC(17) + x where x is exponentially distributed with parameter of 2.

* * * * * INFORMATIONAL ANALYSIS * * * * *

The INFOrmation command may be used to display two measures of the relative importance of each datum (or statistical constraint) with respect to estimating the adjustable parameters of the current model. These information measures are calculated from the partials matrix a, the transpose of the partials matrix a', and the diagonal weight matrix w with w(I,I) equal to the weight of the Ith datum. The matrix A is defined as A = a'*w*a and the matrix invA as the inverse of A. Data analyzed for information content are arranged by

component and w(I,I) is not necessarily equal to W(I). Both information measures are functions only of a and w and not necessarily dependent upon observed data values. Simulated data may therefore be used for INFOrmational analysis.

The first information measure displayed is the "information content" Z(I) and is calculated as Z(I) = w(I,I)*z(I,I) where the z matrix is calculated as z = a*invA*a'. Component number, T value and Z(I) are displayed for each weighted datum and statistical constraint.

The second information measure displayed is H(I) and is calculated as H(I) = log2(detA/detV(I)) = log2(detA) - log2(detV(I)), where detA is the determinant of the A matrix calculated with all the weighted data, detV(I) is the determinant of the A matrix calculated without datum I, and log2(x) is the base 2 logarithm of x. Component number, T value, log2(detV(I)) and H(I) are displayed for each weighted datum and statistical constraint.

Use of the INFOrmation command requires adjustable parameters and prior calculation of the RESIduals vector. An ITERation or a SOLV, PART, RESI sequence must therefore be performed before INFO. The syntax for the INFOrmation command is:

INFO Display informational analysis for current model.

Example:

In the following example comments describing commands and output have been added, in lower-case letters, on the right.

```
$ COPY/LOG INFO.DEK FOR003.DAT                                  copy file to unit 3
%COPY-S-COPIED, DR:[EG]INFO.DEK;1 copied to DR:[EG]FOR003.DAT;3 (1 block)
$ RUN CON27                                                     run consam

*** WELCOME TO CONVERSATIONAL SAAM27 - 20DEC82***

> P DECK                                                        display edit buffer
* DECK BEING READ INTO EDIT BUFFER

   1: A SAAM27
   2: H PAR
   3:    IC(1)    1.
   4:    L(0,1)   .1          0.          1.                    adjustable parameter
   5: H DAT
   6: 101
   7:             0                                            simulated data
   8: 2           2                       9                     requests
```

```
> DECK                                                    create KOMN
* DECK BEING PROCESSED

***ALL WEIGHTS=1.*2*
PRE-PROCESSING TIME :       2.650 SECS
> SOLV                                                    solve for QC's
*** MODEL CODE  2 SOLUTION
SOLUTION TIME :       0.130 SECS
> PRIN Q(1)
-------------------------------------------------------------------------
*** NAME :     1
CURRENT KOMN
 #   COMP TC  CATEGORY       T            QC           QO          QC/QO
  1    1   0  F ( 1)     0.000E+00    1.000E+00    0.000E+00         ****
  2    1   0  F ( 1)     2.000E+00    8.187E-01    0.000E+00         ****
  3    1   0  F ( 1)     4.000E+00    6.703E-01    0.000E+00         ****
  4    1   0  F ( 1)     6.000E+00    5.488E-01    0.000E+00         ****
  5    1   0  F ( 1)     8.000E+00    4.493E-01    0.000E+00         ****
  6    1   0  F ( 1)     1.000E+01    3.679E-01    0.000E+00         ****
  7    1   0  F ( 1)     1.200E+01    3.012E-01    0.000E+00         ****
  8    1   0  F ( 1)     1.400E+01    2.466E-01    0.000E+00         ****
  9    1   0  F ( 1)     1.600E+01    2.019E-01    0.000E+00         ****
 10    1   0  F ( 1)     1.800E+01    1.653E-01    0.000E+00         ****
> PART                                                    calculate partials
* PARTIALS ESTIMATED
PARTIALS TIME :       0.070 SECS
> RESI                                                    calculate residuals
CORRECTION VECTOR TIME :       0.050 SECS
> INFO                                                    request information
                                                           analysis
COMP.        T        INFO.CONTENT


   1  0.0000000E+00  0.0000000E+00                        this section of
   1  2.0000000E+00  2.9581564E-02                         display includes:
   1  4.0000000E+00  7.9160757E-02
   1  6.0000000E+00  1.1915118E-01                          1 component number
   1  8.0000000E+00  1.4170836E-01                          2 time value (T)
   1  1.0000000E+01  1.4812295E-01                          3 Z(I)
   1  1.2000000E+01  1.4269145E-01
   1  1.4000000E+01  1.2993009E-01                         for each datum
   1  1.6000000E+01  1.1352986E-01
   1  1.8000000E+01  9.6123844E-02
```

```
DETERMINANT OF A-MATRIX(LOG BASE 2)= 6.49924E+00          log2(detA)
    1  0.0000000E+00  6.4992375E+00   0.0000000E+00
    1  2.0000000E+00  6.4559164E+00   4.3321133E-02       this section of
    1  4.0000000E+00  6.3802590E+00   1.1897850E-01        display includes:
    1  6.0000000E+00  6.3162036E+00   1.8303394E-01
    1  8.0000000E+00  6.2787776E+00   2.2045994E-01          1 component number
    1  1.0000000E+01  6.2679553E+00   2.3128223E-01          2 time value (T)
    1  1.2000000E+01  6.2771239E+00   2.2211361E-01          3 log2(detV)
    1  1.4000000E+01  6.2984409E+00   2.0079660E-01          4 H(T)
    1  1.6000000E+01  6.3253818E+00   1.7385578E-01
    1  1.8000000E+01  6.3534346E+00   1.4580297E-01        for each datum
> STOP
UPDATE? N
LEAVING CON27
```

Note that in the above example, the datum with the highest information content (by either measure) is at T = 1/L(0,1).

LEAVING CONSAM

Before ending your CONSAM session, you may wish to save KOMN on disk so you will not have to reprocess the source input file at the start of your next session:

WRIT [22-29] — Write KOMN to the logical unit specified. Units 22 through 29 are available for saving KOMN. KOMN will be written to unit 22 if no other unit is specified. This will be file FOR022.DAT on the VAX. KOMN can be a rather large file and it may be more practical to reprocess (DECK) the source input file rather than save KOMN.

The STOP command terminates CONSAM execution and ends a CONSAM session. In response to the STOP command, CONSAM first asks the user whether an UPDAte should be performed. If the user responds with a Y or YES, CONSAM copies the current parameter values into the edit buffer. Whether or not an UPDAte is performed, CONSAM then copies the edit buffer to logical unit 3 (the source input file). This makes sure that any modifications to the source input file are written back to disk. CONSAM then terminates execution and returns the user to the system level. The syntax for the STOP command is:

STOP — Copy the edit buffer to the source input file and leave CONSAM.

After the user is out of CONSAM, the source input file with all changes made within CONSAM (editing, UPDAtes, etc) is available on logical unit 3. This will normally be file FOR003.DAT on the VAX.

Part 2

SPECIAL TOPICS

****
THE SOURCE INPUT FILE
****

* * * * * NOTES ON SAAM NOTATION * * * * *

A complete and detailed description of SAAM options and the format of the source input file (SAAM notation) may be found in the SAAM27 manual (Berman, M. and Weiss, M.F., 1978, DHEW Publication No. (NIH) 78-180). Refer especially to pages III-1 through III-12 for the format of specific SAAM instructions. A brief summary of the essential components of a SAAM deck follows.

A SAAM27 THIS IS A TEST PROBLEM

The A SAAM27 statement must be the first statement in the deck. The title of the problem is optional.

H PAR

The H PAR header statement must precede entries defining parameters and parameter dependencies. These may include:

P(I) any parameter, linear or nonlinear, (I:1-99)

K(I) linear parameter
K(1)-K(25) are multiplying factors for observations in components 1-25
K(26)-K(99) are any linear parameters

L(I,J) fractional rate constant for transfer of material from compartment J to compartment I. The existence of L(I,J) implies a term in the differential equations:

dF(I)/dt = L(I,J)F(J) + ...
dF(J)/dt = -L(I,J)F(J) + ...

I:0,25 I=0 denotes transfer out of the system.
J:1,25

IC(I) Initial conditions in compartment I, (I:1-25)

See page II-9 of the SAAM27 manual for a list of all parameters.
See page III-6 to III-9 of the SAAM27 manual for parameter formats.

H DAT

The H DAT statement must precede entries specifying times at which solution values are to be computed for each component, observed data

values to be fit with a model and explicit equation definitions.

"Field modification" statements, beginning with a "1", may be used to specify or modify entries on succeeding statements:

```
103G(10)    *60.          /100.          FSD=.1
```

```
C A STATEMENT WITH A "C" IN COLUMN 1 IS IGNORED BY SAAM
C OBSERVED DATA:

            0.            .01
            2.4           .356
```

See page III-3 of the SAAM27 manual for a detailed discussion of data specification formats.

G(I) is an algebraic function defined by the user:

```
XG(10)=F(1)**P(1)
```

Example:

Following is a sample SAAM deck, which will fit a set of data to a linear 2-compartmental open model:

```
A SAAM27
H PAR
   IC(1)    100.
   L(2,1)   3.000000E-01  1.000000E-01   9.000000E-01
   L(1,2)   3.500000E-02  1.6666667-02   1.050000E-01
   L(0,1)   1.400000E-02  4.666667E-03   4.200000E-02
H DAT
101                                      FSD=.1
            0.            100.
            5.            61.
            10.           32.
            20.           23.
            30.           18.
            40.           20.
            50.           16.
            60.           17.
101
            0.
2           2.5                          24.
```

* * * * * Y AND Y- * * * * *

Y's and Y-'s can be used to exclude sections of the deck in both SAAM and CONSAM. A Y in column 1 will cause SAAM to ignore subsequent lines in the deck until a Y- in columns 1 and 2 or the end of the deck is encountered. The lines following a Y- are read in the usual fashion. A leading Y- (i.e., a Y- in the deck before any Y's have been encountered) has no effect. Text may also be included on a Y or Y- card since only the first 2 columns are read.

Example:

```
A SAAM27                          EXAMPLE OF Y/Y- FEATURE
H PAR
   K(51)    .35          .1            2
   P(11)    .02
Y ***  THIS AND SUBSEQUENT LINES ARE IGNORED ***
   P(12)    0
   K(53)    0
Y-***  THIS LINE IS IGNORED BUT SUBSEQUENT LINES ARE READ ***
XG(1)=(K(51)*TH)/(P(11)+TH)
H DAT
101G(1)
            0.0
2           .01                           10
Y- *** THIS HAS NO EFFECT SINCE THE STATUS IS READ (ARE READ) ***
2           .05                           20
Y *** THIS IS LAST Y CARD AND CAUSES REMAINDER OF DECK TO BE IGNORED
2           .5                            10
```

****

EDITING THE SOURCE INPUT FILE

****

The CONSAM editor was designed to facilitate the development and updating of a model in a source input file. It thus incorporates standard line editor features and some CONSAM specific features.

Like most editors, the CONSAM editor protects the source input file by making its changes in a buffer before updating the source input file. The first command to the editor automatically brings the contents of the source file into the buffer. Figure 3 shows schematically the source input file, the edit buffer, KOMN and the commands which move information between these entities.

When you are satisfied with modifications you have made in the edit buffer, the DECK command transfers the contents of the buffer to the source input file and processes this file to create a new KOMN. The STOP and RETAin commands may be used to transfer the edit buffer contents to the source input file without processing that file.

If you want to ignore the changes made in the edit buffer, the RESCue command will refresh the buffer with the contents of the source file. All the other commands listed below operate on the buffer, not the source file. It is also important to realize that the CONSAM commands listed in the other sections of this manual operate on KOMN, not the source input file or the edit buffer.

One powerful editor command provides a direct link between your working model in KOMN and the edit buffer. This is the UPDAte command. In response to this command, CONSAM scans the edit buffer for parameters and updates their values in the buffer with their values in KOMN. This command closes the circle:

EDIT BUFFER -> SOURCE INPUT FILE -> KOMN -> EDIT BUFFER.

Besides the standard editor functions of Inserting, DELEting, CHANging, Altering and Printing, the CONSAM editor has consolidated some of the more commonly required text manipulations into single editor commands: TABS, FREE and FIX. These commands facilitate the editing of the editor buffer but do not affect KOMN.

File and information manipulating commands have no arguments, are shown in Figure 3 and include:

DECK — Copy the edit buffer to the source input file and process the latter file to create a new KOMN. If no changes have been made in the edit buffer, processing of the current source input file will occur.

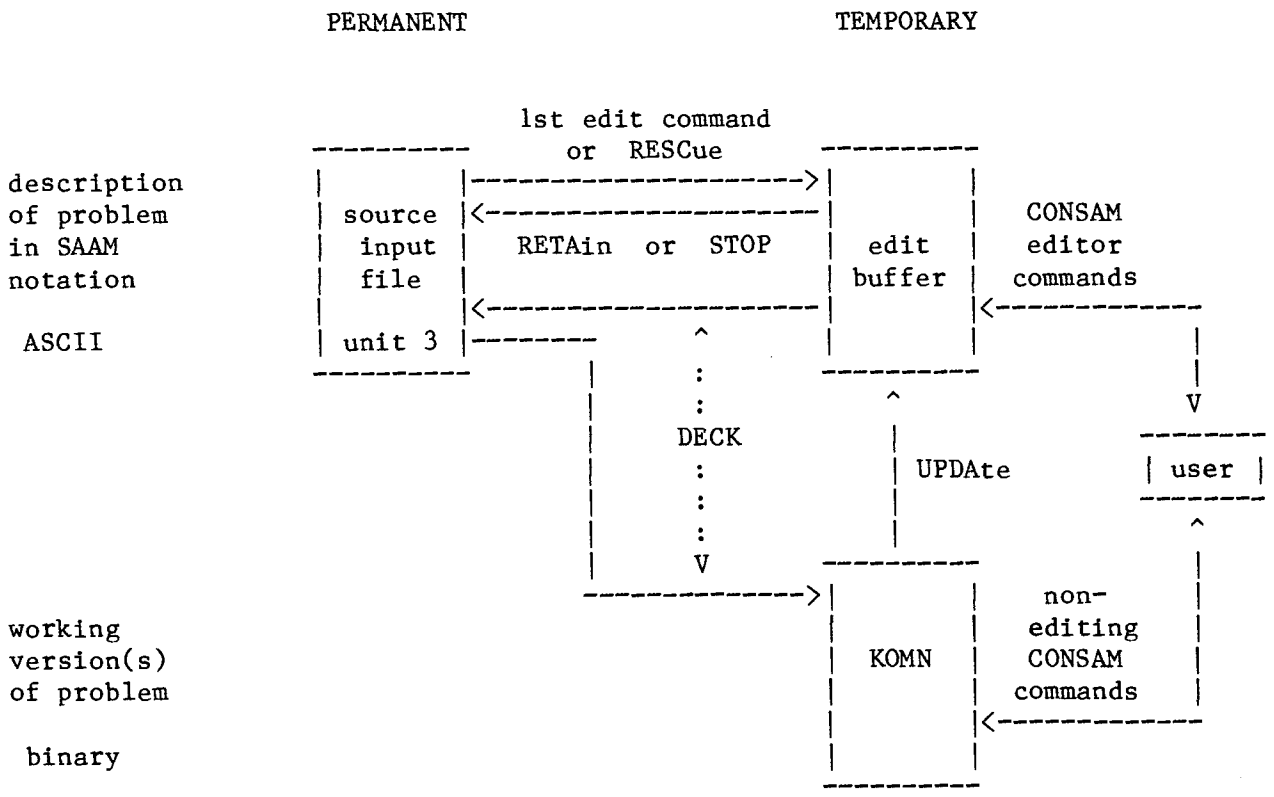


Figure 3

Schematic diagram showing the relationship of the source input file, the edit buffer and KOMN. Note that the user interacts with the edit buffer via the editor commands described in this chapter. Other commands affect KOMN. Changes in the edit buffer are incorporated into KOMN with the DECK command.

RESC — Copy the the source input file into edit buffer. This restores the edit buffer as it was before the last DECK or RETAin.

RETA — Copy the edit buffer into the source input file (see Figure 3). This function is also performed in response to a DECK or STOP command.

UPDA — Replace parameter values in the edit buffer with the current parameter values from KOMN. Limits for adjustable parameters are also replaced. Sections of the edit buffer can be protected from the UPDAte command by using the *, *- (STAR) feature. See page 10.12.

STOP — Copy the edit buffer to the source input file and leave CONSAM (see Chapter 8).

* * * * * DISPLAYING TEXT * * * * *

The CONSAM editor includes a "pointer" which at all times points to a single line in the edit buffer. This line is called the CURRent line. The location of the pointer will depend on the preceding editor commands as several editor commands may move the pointer during processing. The CURRent command displays the line number of the CURRent line. The syntax of the CURRent command is :

CURR — Display current position (line number) of pointer for editing.

The Print command is used to display text from the edit buffer and move the pointer. There are several arguments available with the Print command:

P DECK — Print all lines in the edit buffer. After printing, the line pointer will be at the last line in the edit buffer.

P CURR — Print the current line. The line pointer will not be moved.

P NEXT — Advance position of current line pointer one line and print this line. A <CR> (carriage return with no command) performs the same function as P NEXT.

P LAST — Move the pointer back one line and print this line.

P line[line] — Print the specified line (single argument) or range of lines (two arguments). After this printing, the pointer will be at the last line printed.

Examples:

P 3 33 — Print lines 3 through 33 inclusive. After this command, the pointer will be at line 33.

P 10 — Move the pointer to line 10 and print that line.

P END — Print the last line in the deck.

P STRI — Find the next occurrence of a specified string following current position, move the line pointer to the line containing the string and display that line. If the string is not found, a message will be displayed to that effect and the pointer will be moved to the first line in the edit buffer. CONSAM responds to the P STRI command with the prompt "ENTER STRING:". The user then enters "string>" and CONSAM performs the search.

Example:

ENTER STRING: P(1)> find the next occurrence of the string "P(1)".

P EQUA — Find the next occurence of "=" (equal sign) following the current line pointer position, move the line pointer to the line containing "=" and Print that line. This command is useful to search the source input file for parameter dependency equations. If no "=" is found, a message will be displayed and the pointer will be moved to the first line in the edit buffer.

P — The P command with no argument requests CONSAM to repeat the preceding Print command if it was P LAST, P NEXT, P STRIN or P EQUA and no other commands (including <CR>) have intervened. Otherwise P causes CONSAM to Print the current line (i.e., the same function as P CURR). The P command with no arguments is particularly useful to continue P STRI and P EQUA searches.

Printing of lines relative to the current line may be performed by entering the relative line number. The syntax for this print command is:

integer

where:

integer — is the relative line number. The line printed will be the number of the CURRent line + integer.

Examples:

3	Advance position of current line pointer 3 lines and display this line. If the current line is line 7, line 10 will be displayed. Note that P 3 will always Print line 3.
-12	Move the line pointer back 12 lines and display this line.

* * * * * INSERTING TEXT * * * * *

The Insert command is used to insert one or more lines of text. In response to the Insert command, CONSAM will print the last line before the position of insertion, print a line of column numbers to help align columns and prompt for text to be inserted on a line by line basis. CONSAM will continue to prompt for more lines of text until a blank line is "inserted" (i.e., <CR><CR> terminates insert). CONSAM will then exit from insert mode and echo all lines Inserted. The line pointer will be at the first line after the inserted text. After text has been inserted, all lines following those inserted will have new line numbers. While in insert mode, a "#" in column 1 will reprint the position guides (column numbers). The syntax for the Insert command is:

I [line]

where:

line — is the line number for the first line to be Inserted. If no line number is specified, CONSAM will insert text at the CURRent pointer position.

Example:

I 33	Insert text at line 33. If only one line is Inserted, it will be line 33 and the old line 33 will become line 34.

Although tabulation characters are not allowed in the source input file (except on comment cards), the CONSAM editor allows the use of tab stops to position text in specified columns. A semicolon (;) or colon (:) will be interpreted as a tab, and when insert mode is terminated, all tabs will be converted to the correct number of spaces. On some terminals the TAB key will also function correctly as a TAB, provided the terminal and computer are set correctly. The default tab stops are columns 4, 13, 27, 42 and 56. Tab stops are set or displayed using the TABS command. The syntax of the TABS command is:

TABS [{column[...],DAT,ICC,PAR,PCC}]

where:

column	is the number of a column for which a tab stop is to be set.
[...]	indicates that several columns may be entered separated by spaces.
DAT	requests that tab stops be set for entries under the H DAT header. These stops will be at columns 4, 13, 27, 42 and 56.
ICC	requests that tab stops be set for entries under the H ICC header. These stops will be at columns 2, 4, 9, 13, 27, 42, 44 and 49.
PAR	requests that tab stops be set for entries under the H PAR header. These stops will be at columns 4, 13, 27, 42 and 56.
PCC	requests that tab stops be set for entries under the H PCC header. These stops will be at columns 2, 4, 9, 13, 27, 42, 44 and 49.

If the TABS command is used with no arguments, CONSAM will display the current tab stops.

Examples:

TABS	Display tab stops.
TABS 13 27	Set tabs stops to columns 13 and 27.

* * * * * DELETING TEXT * * * * *

The DELEte command is used to delete one or more lines of text from the edit buffer. The syntax for the DELEte command is:

DELE [line[line]

where:

line — is a line number. If no line numbers are given, CONSAM will DELEte the CURRent line. If one line number is given, CONSAM will DELEte that line. If two line numbers are given, CONSAM will delete the range of lines specified, inclusive. All lines of text after the lines DELEted will have their line numbers decremented appropriately.

Examples:

DELE	Delete the current line.
DELE 10	Delete line 10. The old line 11 will become line 10 and the pointer will be at line 10.
DELE 10 26	Delete lines 10 through 26. The old line 27 will become the new line 10 and the pointer will be at line 10.

* * * * * CHANGING AND ALTERING TEXT * * * * *

The CHANge command is used to replace one character string with another. The syntax for the CHANge command is:

CHAN

CONSAM responds to the CHANge command with the prompt "CHANGE STRING:". The user then enters the string to be replaced and the string which is to replace it. The syntax for the response to the CHANGE STRING: prompt is:

old-string>[>][new-string]>[>]

where:

old-string — is the string to be replaced.

new-string	is the replacement string. If new-string is omitted, the old string will be replaced with a null string, i.e., the old string will be deleted.
[>]	is an optional delimiter used to indicate that all occurences of old-string from the CURRent line to the end of the edit buffer are to be changed.

The search for the string to be replaced begins at the CURRent line and continues to the end of the edit buffer. If the string is not found, CONSAM prints a warning and positions the pointer at the first line of the edit buffer.

Examples of CHANGE STRING responses:

P(21)>P(03)>	Change the next occurrence of "P(21)" to "P(03)".
P(21)>>P(03)>>	Change ALL occurrences of "P(21)" from the CURRent line to the end of the edit buffer to "P(03)".
L(0,4)>>	Delete the next occurrence of "L(0,4)".
L(0,4)>>>	Delete ALL occurrences of "L(0,4)" from the CURRent line to the end of the edit buffer.

The Change command may be used to repeat the last CHANge command if no intervening command (including <CR>) was issued. The Change command is useful for changing more than one occurence of a text string. If the last command was not CHANge, the effect of the Change command is identical to that of the CHANge command, i.e., the user will be prompted for the CHANGE STRING. The syntax of the Change command is:

C

The Alter command accesses a mode that allows direct editing within a line. It also allows you to specify the column number at which the change is to occur. In response to the Alter command, CONSAM will print the current line and prompt the user for the Alteration desired. CONSAM will continue to prompt for Alterations until a blank line is received in response to the alter mode prompt. The syntax for the Alter command is:

A {C,D,I,R} [column]

where:

C — specifies that the alterations desired are Changes of one character string to another. In response to the A C command, CONSAM will return with the "CHANGE STRING:" prompt. The allowable responses to this prompt are identical to those for the CHANge command. See above.

D — specifies that the alterations desired are character Deletions. In this mode, the user spaces across and types a "D" under each character to be deleted. Characters to the right of those Deleted will be shifted to the left. Exit from alter mode will occur in response to any line which contains no D's.

I — specifies that the alterations desired are Insertions of character strings. In this mode, the user spaces across to the desired column and types the string to be inserted. Insertion will begin at the position of the cursor and the old character at that position will be moved ahead. A "$" as the first or last character of the string will be interpreted as a space.

R — specifies that the alterations desired are Replacements of characters. This mode differs from A I in that the old characters are overwritten rather than displaced. In this mode, the user spaces across and types the new characters under those to be replaced. A "$" at the beginning or end of the replacement string will be interpreted as a space.

column — specifies the first column in which alterations are to be made. After each alter mode prompt, CONSAM will position the cursor in the specified column. If no column number is entered, the cursor is positioned at the start of the line.

Examples:

A D 5	Position the cursor at column 5 of the CURRent line. The character in column 5 may be deleted by typing a "D" and then a <CR>. The cursor will then be repositioned at column 5 for further deletions.
A I 42	Position the cursor at column 42 of the CURRent line ready for insertion. This will be repeated until a blank line is encountered.

A R	Position the cursor at column 1 of the CURRent line such that the new characters will overwrite the previous ones.

* * * * * CHANGING THE NATURE OF A PARAMETER * * * * *

The FIX command may be used to remove minimum and maximum values for parameter entries in the edit buffer. This will cause the parameter(s) to be fixed following the next DECK command. The FIX command may also be used to overwrite the value of a fixed parameter in the edit buffer. This will cause the parameter to take that value following the next DECK command. The line of the edit buffer affected will be that with the first occurrence of the specified parameter. Sections of the edit buffer may be protected from the FIX command by using the *, *- (STAR) feature. See page 10.12 for details. The syntax for the FIX command is:

FIX {parameter [value],ALL}

where:

parameter	is the name of a specific parameter (e.g., L(2,1)) or a global parameter name specifying all parameters of a given type (e.g., P(I) or L(I,J)). Single index wild cards such as L(I,3) or S(2,J) are not allowed.
ALL	specifies that all parameter entries in the edit buffer are to be FIXed.
value	is a number (parameter value). This value will be written to columns 13-25 of the appropriate line(s) in the edit buffer.

The FREE command may be used to write minimum and maximum values for parameter entries in the edit buffer. This will cause the parameter(s) to be adjustable following the next DECK command. Sections of the edit buffer may be protected from the FREE command by using the *, *- (STAR) feature. See page 10.12 for details. The syntax for the FREE command is:

FREE {parameter [value [value]],ALL}

where:

parameter	is the name of a specific parameter (e.g., L(2,1)) or a global parameter name specifying all parameters of

a given type (e.g., P(I) or L(I,J)). Single index wild cards such as L(I,3) or S(2,J) are not allowed.

ALL — specifies that all parameter entries in the edit buffer are to be FREEd.

value — is a number. The interpretation given numeric arguments will depend on the number of arguments specified and the values of the arguments. If two values are specified, the lower value will be interpreted as the lower limit and the greater value as the upper limit. If a single positive argument is entered, it will be interpreted as the upper limit and the lower limit will be set to 0. If a single negative argument is entered, it will be interpreted as the lower limit and the upper limit will be set to 0. If no numeric arguments are entered, the lower limit will be set to (parameter value)/3. and the upper limit to (parameter value)*3. For all these possibilities, the lower and upper limits will be written to columns 27-40 and 42-55 respectively of the appropriate line(s) in the edit buffer.

Examples:

FIX L(2,1)	Remove minimum and maximum values for L(2,1) in the edit buffer.
FIX K(25) .001	Set value of K(25) to .001 in the edit buffer. Remove any limits.
FIX P(I)	FIX all P parameters in the edit buffer.
FIX ALL	Make all parameters FIXed by removing any upper or lower limits.
FREE L(0,4) .1 1.	Set minimum value for L(0,4) to .1 and maximum value to 1.
FREE L(I,J) 0. 1.	Set minimum value of 0. and maximum value of 1. for all L parameters.
FREE K(40)	Set minimum for K(40) to .33*K(40) and maximum to 3*K(40).
FREE P(1) -10.	Set minimum for P(1) to -10 and maximum to 0.
FREE P(3) 100.	Set maximum for P(3) to 100. and minimum to 0.

FREE ALL	Transform all parameters specified as fixed or adjustable into adjustable parameters by generating values for lower and upper limits in the edit buffer as .33*par. value and 3*par. value, respectively.

* * * * * STAR AND STAR-MINUS * * * * *

The FIX, FREE and UPDAte commands scan the whole deck while performing their respective operations. A range of lines may be protected from their effects by using * (star) and *- (star-minus) lines in the edit buffer to delimit the protected range. A * in column 1 will protect subsequent lines in the edit buffer until a *- in columns 1 and 2 is encountered. A leading *- (i.e., a *- in the edit buffer before any *'s have been encountered) has no effect. Text may also be included on a * or *- card since only the first 2 columns are read. SAAM ignores cards with a * in column 1.

SAVING MODELS AND SOLUTIONS

The CONSAM user works with 2 major entities:

1. The source input file contains a description of your problem in SAAM notation. This file can be created before running CONSAM by using one of the system editors or within CONSAM by using the CONSAM editor (Chapter 10). See the SAAM Manual (Reference 1) for information on setting up your problem in SAAM notation.

2. KOMN (common) is the working version of your problem. KOMN is created by CONSAM from a source input file in response to the DECK command and is stored in computer memory (CORE).

These model descriptions may be stored in two ways:

1. On DISK. Files stored on disk are permanent and will be available for your next session on the computer. The source input file is automatically stored on disk when you leave CONSAM.

2. In CORE. While you are working, copies of these files or parts of them are stored in CORE memory for fast access. This information is temporary and will be lost if you leave CONSAM, CONSAM leaves you or the computer "crashes".

Several commands allow the user to save all or part of KOMN and later recover this information. These commands are shown schematically in Figure 3 and are the subject of this chapter.

* * * * * SAVING THE MODEL STRUCTURE (KOMN) * * * * *

KOMN may be permanently saved on disk using the WRITe command and recovered using the READ command:

WRIT [unit] — The WRITe command saves a copy of the current KOMN on disk in the file specified by the unit number. This file is referred to as SAVED KOMN and contains the formatted problem, partials, and the current solution. The unit number must be between 22 and 29, inclusive. If no unit number is specified, KOMN is written to unit 22. SAVED KOMN is a large file.

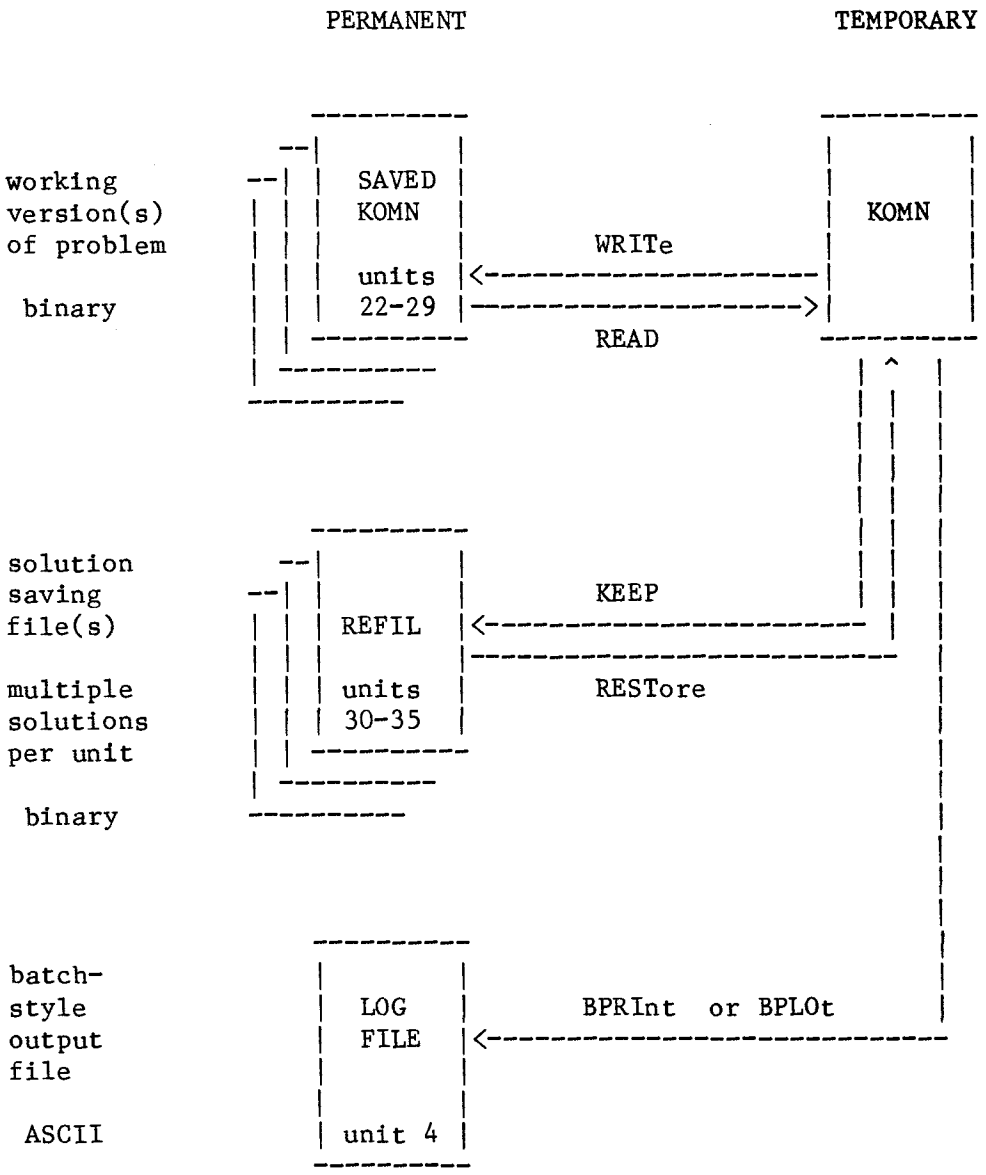


Figure 4

Schematic diagram of the relationship between KOMN and the various model, solution and information saving utilities. Boxes represent entities used in CONSAM operation. Lines reflect information flow between entities and are labeled with the CONSAM commands affecting this information flow. Note that the various units contain different subsets of the information contained in KOMN.

READ [unit] — The read command reads in a copy of KOMN previously saved on disk with the WRITe command (SAVED KOMN). The user may then resume a CONSAM session picking up where the copy of KOMN was WRITten. After READ, you may proceed to SOLVE. It is not necessary for CONSAM to reformat the problem unless specifically desired. SAVED KOMN must have been WRITten to disk before it can be READ.

The TITLe and KOMN commands display information about KOMN. These commands are useful after a READ to verify that you have read the SAVED KOMN that you wanted. The syntaxes for these commands are:

TITL — Display title of problem read from the A SAAM27 card during source input file processing.

KOMN [D] — Display information on KOMN: creation date and operation used to generate it. The optional D (Detail) argument requests a more detailed display of information: the number of parameters, number of data points, etc.

* * * * * SAVING SOLUTIONS * * * * *

Several solutions (parameter values and QC's) for the same model may be saved on disk in a single file. This file is called REFIL. To save solutions, you must first create a new REFIL using the NEW command or open for access an existing REFIL using the OLD command. Each time the deck is changed and the DECK command executed, a new REFIL may be required. CONSAM does consistency checks to make sure that the solutions saved in REFIL are compatible with the current KOMN. Only one REFIL may be open at a time. The syntaxes for the NEW and OLD commands are:

NEW [unit] — Set up a new file (REFIL) for saving solutions.

OLD [unit] — Open a preexisting file for solution access or saving.

where:

unit — is the number of the unit to be opened. The unit specified must be between 30 through 35 inclusive. If no unit is specified, CONSAM will attempt to access unit 30.

The UNITs command may be used to display the unit number of the currently open REFIL. The number of the last unit accessed with a READ or WRITE command will also be displayed, but only if REFIL is open. The syntax for the UNITs command is:

UNIT	Display current unit(s) for saving solutions (and KOMN).

After REFIL has been created or opened, solutions may be saved using the KEEP command. Information saved includes all QC and parameter values for a solution. Many solutions of a given problem may be saved in a single REFIL. They are assigned RECORD numbers, and may be RESTored in any order (see below). Solutions are always stored sequentially. For example, if there are 4 solutions in REFIL, the KEEP command will result in the current solution being stored as solution number 5. The syntax for the KEEP command is:

KEEP	Save the current solution in REFIL.

Solutions stored in REFIL may be restored to KOMN using the RESTore command. This will overwrite the solution and parameters currently in KOMN. The syntax for the RESTore command is:

REST n	Restore the specified solution into KOMN.

where:

n	is a number specifying which solution is to be RESTored. If n is positive, CONSAM will treat it as an absolute solution number and attempt to restore that solution. If n is zero or negetive, CONSAM will treat it as a relative solution number and restore the nth previous solution.

Examples:

REST 3	Copy solution information stored in record (solution) 3 to KOMN.
REST -3	Restore 3rd previous solution. If there are 5 stored solutions, CONSAM will restore solution number 2.
REST 0	Restore the last solution saved.

It is not necessary to RESTore a solution record to examine its contents. Stored parameter values may be displayed using the RELIst command and the PRINt and PLOT commands permit you to specify a REFIL record for which you want QC values printed or plotted.

RELI n — Display parameter values for current KOMN and for the specified REFIL solution. As with the RESTore command, a positive n is interpreted as an absolute solution number and a non-positive n is interpreted as a solution number relative to the last stored solution.

Examples:

RELI 3	Display parameter values for current solution and for solution 3.
RELI -1	Display parameter values for current solution and for the next to last stored solution.

PRIN Q(component[,...]) n[,...]
PLOT Q(component[,...]) n[,...]

PRINt or PLOT the specified data component(s) for the REFIL solution(s) specified by n. Up to 5 components and up to 5 solutions may be displayed with one command. For each solution, a positive n is interpreted as an absolute solution number and a non-positive n is interpreted as a solution number relative to the last stored solution. A special solution designation (n) is the letter C which requests that the current KOMN solution is to be displayed. See Chapter 12 for more information on the PRINt and PLOT commands.

Examples:

PRIN Q(3) 1,2	Print the data in component 3 for saved solutions 1 and 2.
PLOT Q(1,2) C,0	Plot data in components 1 and 2 for the current solution and the last saved solution.

CONSAM contains commands to manipulate the solutions saved in REFIL. The RECOrds command may be used to display information about the records in REFIL, the COUNt command may be used to reset the number of solutions in REFIL and the MOVE command may be used to reposition a specific record in REFIL:

RECO [D]	The RECOrd command requests that the number of solutions currently saved in REFIL be displayed. The D (Detail) argument requests a more Detailed display of information.
COUN n	The COUNt command resets the solution counter to the specified number of solutions. The argument n must be a number between 0 and the number of REFIL records. The number of available REFIL records will be reset to n. All solutions after n will be lost and the next record saved will be number n+1.
MOVE m>n	The MOVE command allows the user to reorder the solutions in REFIL. The arguments m and n must be numbers between 1 and the number of REFIL records. The solution in position m will be moved to position n and the solution previously at n will be lost.

Examples:

COUN 3	Reset solution counter to 3. Next record saved will be #4.
MOVE 10>2	Move REFIL solution record 10 to position 2, overwriting 2.

* * * * * AUTOMATIC SOLUTION SAVING * * * * *

If the STORe switch is ON, solutions will automatically be kept in REFIL after each SOLVe or ITERate command:

SWIT STOR [ON]	Save solution in REFIL after each SOLVe or ITERation.
SWIT STOR O[FF]	Save solution in REFIL only in response to a KEEP.

* * * * * GENERATING BATCH STYLE OUTPUT * * * * *

The BPRInt and BPLOt commands allow the user to send SAAM style data display to the LOG FILE (logical unit 4). This file may then be printed to give the user a permanent record of results similar to that obtained from a batch SAAM run. The BPRInt and BPLOt output will be controlled by the information on card 4, just as in a batch run. The BPRInt command also causes a NEW DECK to be written to unit 9 if card 4 has a "1" in column 9. See the SAAM Manual for details. If both BPRInt and BPLOt are to be used, BPRInt should precede BPLOt as BPRInt will overwrite the current contents of the LOG FILE.

BPRI	Send batch type output (more detailed info on solution) to logical unit 4.
BPLO	Send batch (teletype) plots to logical unit 4.

* * * * * THE EX DECK COMMAND * * * * *

The EX DECK command will execute the lines in the edit buffer as if they were CONSAM commands. On completion the edit buffer will contain the values of the parameters requested, together with the standard deviation of any that are adjustable. EX DECK therefore allows you to obtain specific information from a CONSAM run and store information from different runs in a uniform format.

To use EX DECK it is necessary to first save KOMN using the WRITE command. A new input file (called the EXecutable DECK), containing instructions and a list of the parameters for which values are desired, is then brought into the edit buffer. This file, in addition to the standard parameters, can contain quantities such as LI, TR and FCR (see Chapter 15). Note that global symbols (e.g., U(I) or R(I,J)) do not work in EX DECK and each parameter must be requested separately (e.g., U(2) or R(4,3). Sections of the EXecutable DECK enclosed by Y and Y- are ignored, in the same way as in a regular source input file (Chapter 9).

In the following example comments describing the output have been added, in lower-case letters, on the right.

Example:

```
$ COPY/LOG EXDECK.EG FOR003.DAT
%COPY-S-COPIED, DR:[EG]EXDECK.EG;1 copied to DR:[EG]FOR003.DAT;2 (30 records)
$ RUN CON27                                                  this section enters
                                                             the problem, and writes
*** WELCOME TO CONVERSATIONAL SAAM27 - 20DEC82***            it to FOR022.DAT

> P 1 30
* DECK BEING READ INTO EDIT BUFFER

   1: A SAAM27          EX DECK EXAMPLE
   2: 4  2    1 1
   3: H PAR
   4:    K(1)      4.999973E-01  4.866667E-01   6.500000E-01
   5:    IC(1)    1
   6:    L(2,1)    6.000000E-01  1.500000E-01   1.350000E+00
   7:    L(1,2)    1.000015E-02  6.666666E-03   6.000000E-02
   8:    L(0,2)    1.000000E-01  6.666666E-02   6.000000E-01
   9: H STE
  10:    M(1)     10
  11:    U(2)      5.999910E+01                 100
  12: H DAT
  13: 101                                       FSD=.1
  14:   1          0.000000E+00   5.000000E-01
  15:   1          2.000000E+00   1.581768E-01
  16:   1          4.000000E+00   5.218978E-02
  17:   1          6.000000E+00   1.787549E-02
  18:   1          8.000000E+00   8.847202E-03
  19:   1          1.000000E+01   5.634342E-03
  20:   1          1.200000E+01   3.899792E-03
  21:   1          1.400000E+01   3.138680E-03
  22:   1          1.600000E+01   2.548574E-03
  23:   1          1.800000E+01   1.898147E-03
  24:   1          2.000000E+01   1.698100E-03
  25:   1          2.200000E+01   1.405158E-03
  26:   1          2.400000E+01   1.115260E-03
  27:   1          2.600000E+01   8.883386E-04
  28:   1          2.800000E+01   7.505594E-04
  29:   1          3.000000E+01   6.106103E-04
  30: C ************ END OF SOURCE INPUT FILE ***************

> DECK
* DECK BEING PROCESSED
PRE-PROCESSING TIME :      3.460 SECS
```

```
> WRITE
* REWEIGHTING ALL POINTS *  62
* KOMN WRITTEN TO DEVICE 22
KOMN PROCESSING TIME :      1.050 SECS
> STOP
UPDATE? N
LEAVING CON27
$
$ COPY/LOG EX.DEK FOR003.DAT
%COPY-S-COPIED, DR:[EG]EX.DEK;1 copied to DR:[EG]FOR003.DAT;4 (23 records)
$ RUN CON27

*** WELCOME TO CONVERSATIONAL SAAM27 - 20DEC82***
> P 1 25
* DECK BEING READ INTO EDIT BUFFER

   1: C THS IS THE EXECUTABLE DECK
   2: C IT CONTAINS INSTRUCTIONS AND PARAMETERS
   3:   READ 22
   4:   SOLV
   5:   ITER
   6:   SS
   7:   M(1)
   8:   M(2)
   9:   U(2)
  10:   CALC LI
  11:   LI(1,2)
  12:   LI(2,1)
  13:   CALC FCR
  14:   FCR(1,1)
  15:   FCR(2,2)
  16:   L(2,1)
  17:   L(1,2)
  18:   L(0,2)
  19: Y
  20:  M(1)
  21: Y-
  22:  M(2)
  23:  COR(1,2)

> EX DECK
* KOMN READ FROM DEVICE 22
TITLE: A SAAM27          EX DECK EXAMPLE
CREATION DATE:  6-APR-83
STATUS:  CORRECTION VECTOR (UNUSED)
SYSTEM IDENT:      4752868
KOMN PROCESSING TIME :      1.340 SECS
```

this section enters the executable deck into the buffer area.

the EX DECK command reading from FOR022.DAT

```
*** MODEL CODE 10 SOLUTION                                    solving
SOLUTION TIME :      1.520 SECS
* PARTIALS ESTIMATED                                          iterating
* CORRECTION VECTOR ESTIMATED

CONVERGENCE MEASURES
 IMPROVEMENT IN SUM OF SQUARES =   7.71(%)
 FINAL VALUE OF CONAB =   1.202E+00
 LARGEST CHANGE (   2.67 %) WAS IN PAR( 1, 2)

* CORRECTION VECTOR ESTIMATED

CONVERGENCE MEASURES
 IMPROVEMENT IN SUM OF SQUARES =   0.10(%)
 FINAL VALUE OF CONAB =   9.874E+00
 LARGEST CHANGE (   0.24 %) WAS IN PAR( 1, 2)

ITERATION TIME :     10.320 SECS

DISTRIBUTION OF SQUARES
 COMP  SUM OF SQUARES
   1    2.8463E-08
  *  2.8463E-08
   6:   SS          2.846300E-08                    SS are entered on line 6
  *  1.0000E+01
   7:   M(1)        1.000000E+01                    M(1) is entered on line 7
  *  5.9096E+02
   8:   M(2)        5.909643E+02                    M(2) is entered on line 8
  *  5.9980E+01
   9:   U(2)        5.997973E+01                    U(2) is entered on line 9
*** MODEL CODE 10 SOLUTION                          LI is calculated
SOLUTION TIME :      1.310 SECS
LI(  1, 2) =   1.6672E-01                           The LI's are entered
  11:   LI(1,2)    1.667230E-01
LI(  2, 1) =   9.8527E+00
  12:   LI(2,1)    9.852734E+00
FCR( 1, 1) =   5.5258E-01                           The FCR's are entered
  14:   FCR(1,1)   5.525784E-01
FCR( 2, 2) =   1.0149E-01
  15:   FCR(2,2)   1.014947E-01
  *  6.0865E-01 A
* VALUES MAY NOT RELATE TO CURRENT PARAMETERS       L's, min, max and SD's
* L ( 2, 1)    6.087E-01      SD( 1)   8.913E-03
  16:   L(2,1)     6.086521E-01  1.500000E-01   1.350000E+00        8.913E-03
  *  1.0299E-02 A
* VALUES MAY NOT RELATE TO CURRENT PARAMETERS
```

```
* L ( 1, 2)    1.030E-02      SD( 2)   4.533E-04
  17:   L(1,2)     1.029930E-02  6.666666E-03   6.000000E-02        4.533E-04
  *  1.0149E-01 A
* VALUES MAY NOT RELATE TO CURRENT PARAMETERS
* L ( 0, 2)    1.015E-01      SD( 3)   2.056E-03
  18:   L(0,2)     1.014947E-01  6.666666E-02   6.000000E-01        2.056E-03
  *  5.9096E+02
  22:  M(2)        5.909643E+02                          n.b. Y/Y- are ignored;
* VALUES MAY NOT RELATE TO CURRENT PARAMETERS           M(2) and correlation
  *  3.8299E-01                                         value are entered
  23:  COR(1,2)    3.829872E-01
                                                        This completes the
                                                        EX DECK command.
> P 1 30

   1: C THS IS THE EXECUTABLE FILE
   2: C IT CONTAINS INSTRUCTIONS AND PARAMETERS
   3:   READ 22
   4:   SOLV
   5:   ITER
   6:   SS          2.846300E-08
   7:   M(1)        1.000000E+01                       The edit buffer now
   8:   M(2)        5.909643E+02                       contains the values
   9:   U(2)        5.997973E+01                       of the paramters for
  10:   CALC LI                                        the problem stored in
  11:   LI(1,2)     1.667230E-01                       FOR022.DAT
  12:   LI(2,1)     9.852734E+00
  13:   CALC FCR
  14:   FCR(1,1)    5.525784E-01
  15:   FCR(2,2)    1.014947E-01
  16:   L(2,1)      6.086521E-01  1.500000E-01   1.350000E+00        8.913E-03
  17:   L(1,2)      1.029930E-02  6.666666E-03   6.000000E-02        4.533E-04
  18:   L(0,2)      1.014947E-01  6.666666E-02   6.000000E-01        2.056E-03
  19: Y
  20:  M(1)                                             This value is not
  21: Y-                                                entered because it was
  22:  M(2)        5.909643E+02                         in Y/Y-
  23:  COR(1,2)    3.829872E-01

> STOP
UPDATE? N
* EDIT BUFFER BEING WRITTEN INTO DECK                  The buffer is now
LEAVING CON27                                          written to FOR003.DAT
```

ADVANCED PLOTTING AND PRINTING

To PLOT observed and calculated data on a Tektronix 4020 series graphics terminal, one needs only to type

PLOT Q(component)

where:

component is the number of the data component to be plotted.

If you are not using a Tektronix 4020 series terminal, refer to the section on DEVIces below. In response to the PLOT command, CONSAM will automatically scale a semilog graph and plot the data. Up to 5 components may be plotted with one PLOT command. All data are plotted on the same graph. CONSAM contains several features which allow plotting on non-4020 terminals, overriding the default graph characteristics and plotting different data sets on different axes. This chapter describes these features and the plotting of custom-made graphs. Also described in this chapter are the PRINt and ERASe commands, which use the same syntax as the PLOT command. The arguments for a PLOT, PRINt or ERASe command are retained by CONSAM and the use of any of these commands without arguments will cause the PLOTting, PRINting or ERASure of the data specified in the preceding PLOT, PRINt or ERASe command.

* * * * * PLOTTING SOLUTIONS * * * * *

There are five syntaxes for the PLOT command:

PLOT data-type(name[*factor][#lint][,...]) [solution[,...]]

PLOT the specified data. Use the specified line types for display of the specified data segments or components.

PLOT data-type(name[*factor][,...]) [solution[#lint][,...]]

PLOT the specified data. Use the specified line types for display of the specified REFIL solutions.

PLOT {P,S}(name[*factor][#lint][,...]) parameter[,...]

PLOT partials or sensitivities. Use the specified line types for display of the specified data segments or components.

PLOT {P,S}(name[*factor][,...]) parameter[#lint][,...]

PLOT partials or sensitivities. Use the specified line types for display of the specified REFIL solutions.

PLOT

PLOT the data, partials or sensitivities requested in the last PLOT, PRINt or ERASe command.

where:

data-type — is the type of data to be PLOTted. Currently available data-types are:

- QO — observed data values. Observed data are plotted on graphics terminals as open symbols.
- QC — calculated data values. Calculated data are plotted on graphics treminals as continuous lines.
- Q — both observed and calculated data values.
- RES — the residual QO-QC.
- FSE — fractional standard error
- R — the ratio QC/QO.
- W — statistical weights assigned to the data. The QO's are plotted as open symbols and the W's as a continuous line.
- TH — Theta: second independent variable

name — is either the number of a data component or the name of a data segment (see Chapter 13 on names and naming). Up to 5 components and/or names may be listed in one PLOT command.

factor	is the scale factor by which the data in the specified component or name is to be multiplied by before plotting.
lint	is a number between 1 and 5 (1 or 2 for devices 33 and 1200) which specifies the type of line and symbol to be used for plotting the specified component or name. See pages 12.5-12.6 for details.
solution	is the number of a REFIL solution for which data is to be plotted. See Chapter 11 for details on how solution numbers are interpreted.
P	specifies that Partials are to be plotted.
S	specifies that Sensitivities are to be plotted.
parameter	is the name of a specific adjustable parameter with respect to which the partials or sensitivities are desired.

Examples:

PLOT Q(FREE)	Plot observed and calculated points in data segment FREE for solution currently in KOMN.
PLOT QC(20) 4,C	Plot calculated points for component 20 from record 4 and current solution in KOMN.
PLOT QO(C6#3)	Plot observed values from data segment C6 using line type 3.
PLOT QC(IN*1.5)	Multiply calculated values for data segment IN by scale factor of 1.5 and plot.

PLOT Q(20,C6#3,FREE*2,Z*1.5#5) 1,3,9,C

Plot calculated and observed points for data segments 20, C6, FREE and Z for records 1, 3, and 9 and for the current solution. Use line type 3 for C6 and line type 5 for Z. Multiply points in FREE by 2 and points in Z by 1.5 before plotting.

PLOT S(OP,1) L(4,3),P(10)

Plot sensitivities of names OP and 1 with respect to parameters L(4,3) and P(10).

PLOT P(ABC) L(1,2)#3,P(2)	Plot partials of ABC with respect to L(1,2) with line type 3 and with respect to P(2).
PLOT W(3)	Plot weights for data in component 3.
PLOT R(4)	Plot ratios of QC to QO for component 4.
PLOT TH(23)	Plot theta for component 23.
PLOT S(1) L(0,1)	Plot sensitivity curve for component 1 with respect to L(0,1).
PLOT P(1) L(0,1)	Plot partial derivatives of component 1 with respect to L(0,1).
PLOT	Plot data specified in last PLOT, PRINt, or ERASe command.
PLOT RES(23)	Plot the residuals, QO-QC, for component 23.
PLOT FSE(21)	Plot the fractional standard error, (QO-QC)/QO for component 21.

* * * * * PROGRAMMED KEYS * * * * *

After a PLOT command on a Tektronix 4020 series terminal, the special function keys F2 to F5 and S1 to S5 at the top of the keyboard and PT at the extreme right end of the keyboard are programmed to perform special functions which are useful for labeling, examining and copying the graph(s). F1 refers to striking the key labeled "F1" on the keyboard, S1 refers to Shift-F1, or striking the F1 key while holding down the shift key, etc. The following keys are programmed:

Key	Function	
S1	Erase graph	Erases workspace (graphics area).
S2	Enter label	Characters typed at keyboard will be displayed in the workspace.
F2	Exit label	Characters typed at keyboard will be displayed in monitor area and sent to the computer.

Key	Function	
S3	Enter append	Text coming from the computer will be displayed in the workspace below the graph. This will include the echoing of characters typed at the keyboard.
F3	Exit append	Text coming from the computer will be displayed in the monitor area.
S4	Bottom label	Places a hyphen (-) near the last line which will appear on hard copies.
F4	Vertical label	Moves workspace cursor down one line and left one character, i.e., the next character typed will be directly below the last character typed.
S5	Upper half	Moves workspace cursor so as to display top half of graphics area.
F5	Lower half	Moves workspace cursor so as to display bottom half of graphics area.
PT	Hard copy	Generates hard copy of workspace contents. Requires Tektronix hard copy device.

* * * * * LINE TYPES * * * * *

Five (5) line types are available for plotting on a Tektronix terminal or Zeta plotter. Two (2) line types are available for character plots (DEVIce 33 or 1200). The LINType command is used to display the available line types or to set the line type to be used for the next PLOT. The syntax for the LINType command is:

LINT [{1-5}]

where if no line type is specified, a list of available line types will be displayed.

For Tektronix terminals and ZETA plotters, the following line types are available:

LINT	QC line	QO symbols
1	solid line	triangles
2	dotted line	squares
3	dashed line	inverted triangles
4	broken line	diamonds
5	long broken line	stars

For DEVIces 33 and 1200 (character plots), the following line type are available:

LINT	QC symbol	QO symbol	QC+QO symbol	Unweighted QO symbol
1	+	*	X	N
2	$	8	=	Z

Examples:

LINT	Display key to line and symbol types.
LINT 2	Set symbol or line type for next plot to line type 2.

* * * * * ERASING PLOTS * * * * *

The ERASe command may be used with Tektronix 4020 series terminals to ERASe plotted data. As with the PLOT command, there are five syntaxes for the ERASe command:

ERAS data-type(name[*factor][#lint][,...]) [solution[,...]]

ERASe the specified data. Use the specified line types for the specified data segments or components.

ERAS data-type(name[*factor][,...]) [solution[#lint][,...]]

ERASe the specified data. Use the specified line types with the specified REFIL solutions.

ERAS {P,S}(name[*factor][#lint][,...]) parameter[,...]

ERASe partials or sensitivities. Use the specified line types for the specified data segments or components.

ERAS {P,S}(name[*factor][,...]) parameter[#lint][,...]

ERASe partials or sensitivities. Use the specified line types for the specified REFIL solutions.

ERAS

ERASe the data, partials or sensitivities requested in the last PLOT, PRINt or ERASe command.

where the arguments are identical to those for the PLOT command (see above). Although specifying specific line types does not effect the erasure of lines (e.g., QC data), they are useful for the erasure of QO symbols to make sure the entire symbol is erased.

* * * * * DEVICES * * * * *

The DEVIce command is used to specify the type of terminal you are using. The default DEVIce is 4025. If you are not using a Tektronix 4020 series terminal or a terminal which will respond 4020 style graphics, you must set the DEVIce type before PLOTting. The syntax for the DEVIce command is:

DEVI[=device-type]

where:

device-type is the type of device desired. If no device-type is specified, CONSAM will display the current device-type.

The following device types are currently available in CONSAM:

device-type	device(s) supported
33	Standard, non-graphics teletype or terminal. CONSAM will generate character plots which are 80 characters wide.
230	EAI ZETA plotter (110 or 300 baud).
1200	Terminet 1200 or any non-graphics terminal which can display 118 or more characters per line. CONSAM will generate character plots which are 118 characters wide.
4010	Tektronix 4010 series graphics terminals (4006, 4010, 4012 or 4014) or their equivalent.
4025	Tektronix 4020 series graphics terminals (4025 or 4027) or their equivalent.

Examples:

DEVI	Display the current device type.
DEVI=4025	Set device type to Tektronix 4020 series graphics terminal.

The LIST command with the DEVIce argument may be used to display the available device types supported by CONSAM:

LIST DEVI	List the devices which can be driven by CONSAM and their respective codes (device types).

* * * * * PLOTTING SWITCHES * * * * *

Several switches affect the results of the PLOT command. Refer to Page (1.3) for the syntax of the SWITch command. Remember that the STAT command may be used to inspect the STATus of all switches. The Plotting switches are:

AUTO ON: plots are rescaled unless TSCAle or YSCAle is set manually. Each time the screen is erased, a new scale will be calculated and used unless TSCAle G or YSCAle G has been set explicitly.

OFF: graphs will not be rescaled until you specify TSCAle Clear or YSCAle Clear. Once a scale has been computed it is not recomputed automatically when the screen is erased. This may be useful if you want to compare exact values on hard copy plots.

<DEFAULT=ON>

CONNect ON: QO symbols are connected by a solid line.

OFF: QO symbols are not connected.

<DEFAULT=OFF>

FLOW ON: QC or QO values which are outside the current graphics window are represented by the character 'o' for overflow or 'u' for underflow.

OFF: No labeling of underflow/overflow value.

<DEFAULT=OFF>

FRAMe ON: graphics window is enclosed by rectangle (4010 or 4025 only).

OFF: coordinate axes only.

<DEFAULT=OFF>

HCOPy ON: graphs are automatically expanded to fill one page of hard copy output (4025 only).

OFF: graphs not expanded.

<DEFAULT=OFF>

LABEl ON: in single plot mode, ticks and numbers appear on axes for plots. In multiple mode, axes are drawn.

OFF: in single plot mode, ticks and numbers do not appear. In multiple mode, axes are not drawn.

<DEFAULT=ON>

LINL ON: linear region appears at the bottom of logarithmic plots.

OFF: no linear region appears in logarithmic plots.

<DEFAULT=ON>

LOG ON: plots are displayed semi-logarithmically.

OFF: plots are displayed linearly.

<DEFAULT=ON>

MULT ON: multiple plot mode (see below).

OFF: single plot mode.

<DEFAULT=OFF>

STREtch ON: T-axis is distorted locally if points are very close together.

OFF: no distortion.

<DEFAULT=OFF for 4025, 4010 or 230; ON for 33 or 1200>

* * * * * SCALING GRAPHS * * * * *

If you do not specify a scale, plots will automatically be scaled to the maximum T and Y values of all the points included in the PLOT command. The TSCAle, YSCAle, TRANge, YRANge, CYCLes and SCALe commands allow the user to override the automatic scaling. PLOTting parameters set using these commands are valid for all DEVIce types, are stored on unit 16 and will still be in effect during later CONSAM sessions if access to unit 16 is retained.

The TSCAle and YSCAle commands allow the user to specify the minimum and maximum values for the horizontal and vertical axes respectively. The TRANge and YRANge commands are used to specify the ranges of T and Y values which are to be displayed. If a range is set, only data which have T or Y values within the specified range are displayed. The syntax is the same for all four commands:

TSCA G [{value[value],C}] Set or clear horizontal axis scaling.

YSCA G [{value[value],C}] Set or clear vertical axis scaling.

TRAN G [{value[value],C}]	Set or clear independent variable (T) range. Only data with T values within the range will be displayed.
YRAN G [{value[value],C}]	Set or clear dependent variable (Y) range. Only data with Y values within the range will be displayed.

where:

G — specifies that the range or scale is for General plots.

value — is a number to be used as the minimum or maximum axis value (TSCAle or YSCAle) or the minimum or maximum range limit (TRANge or YRANge). If two values are entered, the lesser number will be used as the minimum and the greater number as the maximum. If one value is entered, it will be taken as the maximum if it is greater than zero and as the minimum if it is less than zero. The other limit will be set equal to 0. If no values are entered, the current minimum and maximum will be displayed.

C — specifies that previously set scale or range is to be cleared. For TSCAle and YSCAle, this will result in automatic scaling of the appropriate axis during the next PLOT (even if the AUTO SWITch is OFF).

Examples:

TSCA G C	Clear specified T axis scale. Determine horizontal scaling automatically for data of next PLOT request.
TSCA G	Display max and min for T axis.
YSCA G 0 100.	Set Y axis min to 0 and max to 100.
TRAN G 3 99	Restrict T range of points plotted to 3<T<99. Do not plot outside this range.
YRAN G C	Clear Y range restrictions.

The CYCLes command is used to set the number of CYCLes to be plotted on semilogarithmic plots. The syntax for the CYCLes command is:

CYCL G [{1-4,C}]

where:

G	specifies that the cycle number is for General plots.
1-4	is the number of cycles to be used on the Y axis of semilog plots. If no cycle number is entered, the current cycle number will be displayed.
C	specifies that the previously set cycle number is to be cleared. This will reset the cycle number to the default value of 2.

Example:

CYCL G 3	Set Y axis on semilog plots to 3 cycles.

The SCALe command may be used to specify scaling factors by which the values to be plotted or printed are to be multiplied. This multiplication will take place before the axes are scaled. The syntax for the SCALe command is:

SCAL G [{t-factor[y-factor],C}]

where:

G	specifies that the scaling is for General plots.
t-factor	is the multiplicative factor by which all T values are to be scaled before display. If no scaling factors are entered, the current scaling factors will be displayed.
y-factor	is the multiplicative factor by which all Y values are to be scaled before display.
C	specifies that previously set scaling is to be cleared.

Example:

SCAL G 60 .77	When plotting or printing any data, multiply T values by 60 (e.g., to convert hr to min) and multiply Y values by .77

* * * * * MULTIPLE PLOT MODE * * * * *

On Tektronix terminals and ZETA plotters, multiple plot mode may be used to plot different data components and named data segments on separate graphs using different axes and plotting parameters. On the Tektronix 4025, multiple plot mode makes use of the full graphics space, which is about twice the screen height in the vertical dimension. On the 4010 series, only the visible screen is used. As many plots as the user wishes may be displayed in this area by shortening the axes of the graphs and offsetting their origins.

To PLOT in multiple plot mode, the MULT switch must be ON:

SWIT MULT [ON]	Use plotting parameters specified for each component or named data segment rather than those for General plots.

As in single plot mode, the scales will automatically be computed for each component or name plotted. Alternatively, scales and ranges may be set individually for each name in multiple plot mode. These scales apply ONLY in multiple plot mode. In single plot mode, scales and ranges set for "G" are used.

The AXIS command is used to set the size of a graph for a given component or named data segment. The syntax for the AXIS command is:

AXIS name [{t-length y-length,C}]

where:

name	is a data component number or the name of a defined data segment.
t-length	is the length of the T (horizontal) axis measured as fraction of the terminal screen width. If no axis lengths are entered, the current axis lengths for the specified name will be displayed.
y-length	is the length of the Y (vertical) axis measured as fraction of the display area height.
C	specifies that previously set axis lengths are to be cleared. This will result in axis lengths being set to the default values of t-length = 1. and y-length = .5

Example:

AXIS GTP 1. .25	Set length of T axis for name "GTP" to 1 screen width and Y axis height to .25 of graphics space (approximately .5 screen height).

The OFFSet command is used to position a graph for a given component or named data segment. The syntax for the OFFSet command is:

OFFS name [{t-coord y-coord,C}]

where:

name	is a data component number or the name of a defined data segment.
t-coord	is the horizontal coordinate of the lower left corner of the graph. This coordinate is measured as fraction of terminal screen width. If no coordinates are entered, the current OFFSets for the specified name will be displayed.
y-coord	is the vertical coordinate of the lower left corner of the graph. This coordinate is measured as fraction of the display area height.
C	specifies that previously set OFFSets are to be cleared. This will result in OFFSets being set to the default values of 0, 0.

Example:

OFFS A70 0 .5	Offset origin of plots for name "A70" by 0 in T direction and .5 in Y direction (plot in top half of graphics space).

The scaling commands TSCAle, YSCAle, TRANge, YRANge, CYCLes and SCALe have the same format for multiple plot mode as for single plot mode, except that separate scales and ranges may be specified for each component or name. The syntaxes for these commands are:

TSCA name [{value[value],C}]	Set or clear horizontal axis scaling for the specified data.

YSCA name [{value[value],C}]	Set or clear vertical axis scaling for the specified data.
TRAN name [{value[value],C}]	Set or clear independent variable (T) range for the specified data.
YRAN name [{value[value],C}]	Set or clear dependent variable (Y) range for the specified data.
CYCL name [{1-4,C}]	Set or clear cycle number for the specified data.
SCAL name [{t-factor[y-factor],C}]	Set or clear scaling factors for the specified data.

where:

name — is a data component number or the name of a defined data segment.

C — specifies that previously set plotting parameters are to be cleared.

value, t-factor and y-factor — have the same interpretations as in single plot mode (see above).

Examples:

TSCA ABC C	Clear specified T axis scale for name ABC and determine scales automatically next time ABC is plotted.
YSCA 3	Display max and min Y axis values for component 3.
YSCA GLU 0 100.	Set Y axis min to 0 and max to 100 for name GLU.
TRAN U 3 99	Restrict T range of plotted points from data segment U to 3<T<99.

CYCL 2 3	Set Y axis on semilog plots for component 2 to 3 cycles.
SCAL XXX 1. 3.	When plotting or printing data in data segment XXX, multiply T values by 1. (i.e., do not change T values) and multiply Y values by 3.

* * * * * PRINTING SOLUTIONS * * * * *

The PRINt command may be used with any DEVIce to display data in tabular form. For each component or named data segment requested, the display includes: the name under which the data were requested, any TRANge currently in effect and any SCALing factors currently in effect (if not equal to 1.). For each datum, the display includes: datum number, component number, TC block, category, T value and whatever data-type was requested in the PRINt command. Arguments for the PRINt command are similar to those for the PLOT command. Although line type and factor specifications are allowed in the PRINt command, they have no effect on the display. There are therefore three syntaxes for the PRINt command:

PRIN data-type(name[,...]) [solution[,...]]

PRINt the specified data.

PRIN {P,S}(name[,...]) parameter[,...]

PRINt partials or sensitivities.

PRIN

PRINt the data, partials or sensitivities requested in the last PLOT, PRINt or ERASe command.

where the arguments are identical to those for the PLOT command (see above).

Examples:

PRIN Q(1,Z)	Display observed values, calculated values and QC/QO for data segments 1 and Z.
PRIN QO(GORK) 3,5	Print observed values in data segment "GORK" from REFIL solution records 3 and 5.

PRIN QC(A,9)	Print calculated values for names A and 9.
PRIN P(Z) L(0,1)	Print partials of Z with respect to L(0,1).
PRIN S(A,B,9) P(50)	Print sensitivities of A, B and 9 with respect to P(50).
PRIN	Print data specified in last PRINt, PLOT or ERASe command.

****

NAMING DATA SEGMENTS

****

The NAME command allows the user to specify logical collections of data (data segments) which may then be treated as a unit for displaying and unweighting. For example, if data in component 1 during TC(0) and in component 5 during TC(1) form a logical unit, a single NAME can be defined to include both. Information on using names is given in the sections which describe the appropriate commands. See Chapter 12 for information on PLOTting and PRINting named data segments. Pages 12.13 to 12.16 of that chapter describe multiple plot mode and plotting different named data segments on different axes. See page 6.4 for information on UNWEighting and REWEighting data segments during data fitting. NAMEs are preserved between CONSAM sessions on logical unit 16. Up to 25 names may be defined.

* * * * * DEFINING NAMES * * * * *

The syntax for defining a name is:

```
NAME name=C:component[<[TC(index)][,][category]>][,...]
```

where:

name	is the name to be defined. A name may consist of from 1 to to 4 alphanumeric characters but must start with a letter. The special name G is reserved for general plotting information. The special name ALL is reserved for UNWEighting or REWEighting all data.
component	is a data component number (an integer between 0 and 25 inclusive). Up to five components may be listed in the name command and the same component may appear more than once. Components in the list are separated by commas.
index	specifies a TC block within the component for which data is to be associated with the name. Only one TC block may be specified for each component entry.
category	specifies a category of data (e.g., parameter name or function name) within the component to be associated with the name. Only one category may be specified for each component entry.

Examples:

NAME NGF=C:1,C:24	Define the name NGF to contain all data in components 1 and 24.
NAME IGG=C:2<TC(2)>	Define the name IGG to contain all points in component 2 during T-interrupt block 2.
NAME H2O=C:19<G(13)>	Define the name H2O to contain all points in component 19 under category G(13).
NAME PETE=C:3<TC(11),G(99)>	Define the name PETE to contain all data in component 3 during T-interrupt block 11 and under category G(99).
NAME MM=C:1<M(1)>,C:1<M(2)>	Define the name MM to contain all data in component 1 under category M(1) or M(2).

* * * * * DISPLAYING INFORMATION ABOUT NAMES * * * * *

The NAME command may also be used to display information about defined names:

NAME

The NAME command with no arguments prints a list of names which are currently defined.

NAME name[/F]

The NAME command with a single argument will display information about a name if the argument is a defined name (including G or a component number). The /F (FULL) qualifier requests a full description of the name. If the /F qualifier is omitted, CONSAM will display the component(s), TC block(s) and category(s) associated with the name. If the /Full qualifier is included, CONSAM will also display the plotting information associated with the name. For the special name G, CONSAM displays plotting information with or without the /Full qualifier.

Examples:

NAME CAMP	display the constituent components of the name CAMP.

NAME G	display single plot mode scaling information.
NAME CGMP/F	display full description of information associated with the name CGMP, including scaling information for multiple plots.
NAME 5/F	display full description of multiple plot mode scaling information for data component 5.

* * * * * DELETING NAMES * * * * *

The NAME command may also be used to delete defined names. The special name G may not be deleted. The syntax for deleting a name is:

NAME name/DELE

Example:

NAME ARAC/DELE	Delete the name ARAC and associated information from name table. This name will no longer be available unless redefined later.

****

DISPLAYING A MODEL

****

Using the DRAW, LOCAte, REROute and SUBSet commands you can display the current model in KOMN on a Tektronix 4020 series graphics terminal. The DRAW command alone will display the current model as a circle of compartments. A maximum of 12 compartments can be displayed in this circle. To ease interpretation and to display larger models, the LOCAte, REROute, and SUBSet commands permit you to specify the positions and sizes of the compartments, state the pathway by which 2 compartments are to be connected, and designate the compartments to be displayed, thus defining a "model structure". The DRAW command combines the model structure with the current KOMN to produce its display.

Several different model structures can be stored in logical unit 40. These are called SUBSystems and each is assigned a name by the user. Once a subsystem is referred to in any of the above commands, that subsystem becomes the "default subsystem" for subsequent commands until another subsystem is referred to. The subsytem G is special and refers to the general model. Subsystem G must contain all nonnegative compartments.

Negative compartment numbers refer to user definable points. They are used in the REROute command to redirect connections between compartments and to draw figures. Negative compartments are not labeled in the drawing unless specifically requested.

Although 4025 is the default device type, the DRAW and LOCA/CURS commands require that CONSAM know whether the terminal being used can recognize certain commands. Before using any of the commands described in this chapter, the user must therefore set the device type to 4025:

DEVI=4025 Set device type to Tektronix 4020 series terminal.

In response to this command, CONSAM will prompt the terminal to display some information about itself and ask the user a question or two about this information.

* * * * * DRAWING A MODEL * * * * *

The syntax for the DRAW command is:

DRAW[/AUTO][/ERAS][/GRID][/MINU][/NOLA][/NORE] [subsystem]

where:

/AUTO requests that the default (circle) arrangement be used.

/ERAS	requests that the specified structure be erased. Compartment numbers (labels) are not erased.
/GRID	requests that a reference GRID be drawn as an aid in positioning compartments.
/MINU	requests that negative compartment numbers be displayed.
/NOLA	requests that compartment numbers (labels) not be displayed.
/NORE	requests that RERoutes not be used.
subsystem	specifies the subsystem to be drawn. This subsystem will also become the default subsystem. If no subsystem is specified, the default subsystem will be drawn.

Examples:

DRAW	Draw the model using the model structure of the default subsystem.
DRAW/NORE INSU	Draw the model using model structure INSU. Do not RERoute the interconnections between compartments.
DRAW/GRID/MINU LIPO	Draw the model using model structure LIPO. Superimpose the coordinate GRID and label the negative (MINUs) compartments.
DRAW/AUTO G	Draw the model using the general model structure. Position the active compartments AUTOmatically in a circle.
DRAW/NOLA G	Draw the model using the general model structure. Do not label it.
DRAW/ERAS	Erase the model preserving the labelling information.

* * * * * ESTABLISHING A MODEL STRUCTURE * * * * *

The LOCAte command allows the user to examine and set the location and size of a compartment, the position and size of figures associated with it (L(out), IC, UF and K), and the size of the gap separating bidirectional transfers (L(x,y) and L(y,x)).

The X and Y coordinates for a compartment's position and the radius of the compartmental symbol are measured as fractions of the of the graphics area, i.e, the units are the same as those for axes and offsets in multiple plot mode. Coordinates and sizes are retained for each compartment of each subsystem.

The position of a figure attached to a compartment (e.g., the arrow representing an initial condition) is specified as an angle in degrees. This angle corresponds to standard polar coordinates: 0 is to the right and progression is counterclockwise. A negative angle will cause the figure to not be drawn. The size of an IC(I), L(0,I) or UF(I) arrow attached to a compartment is measured as the ratio between the length of the arrow and the radius of the compartment to which it is attached. Angles are retained for each compartment of each subsystem. Relative radii for each type of attached figure are the same for all compartments of a subsystem.

The gap between bidirectional arrows is measured in pixels. A pixel is the smallest representable distance in a graphics display. Only one gap is defined for each subsystem.

The LOCAte command with no arguments or a single argument which is the name of a subsystem may be used to display the drawing attributes for all active compartments in a subsystem. An active compartment is one which is defined in both the current model and in the user defined model structure. To display attributes of inactive compartments use the /ALL qualifier (see below).

LOCA [subsystem]

where:

subsystem — specifies the subsystem for which drawing information is desired. This subsystem will also become the default subsystem. If the subsystem argument is omitted, CONSAM will display LOCAtion information for all acitve compartments of the default subsystem.

The LOCAte command with a compartment number as the first or second argument is used to set or display drawing attributes for a single compartment. You may set attributes for either an active or an inactive compartment. There are two syntaxes for this form of the LOCA command. Note that the second format allows that each of the compartment's attributes can be set using the first letter of the attribute as a label:

```
LOCA [subsystem] compartment [x-coord [y-coord [radius [o-angle [i-angle
                             [u-angle [k-angle]]]]]]]

LOCA [subsystem] compartment [X:x-coord] [Y:y-coord] [R:radius] [O:o- angle]
                             [I:i-angle] [U:u-angle] [K:k-angle]
```

where:

subsystem	is the subsystem in which attributes for the specified compartment are to be set or displayed. This subsysten will also become the default subsystem. If no subsystem is specified, CONSAM will set or display drawing attributes for all active compartments of the default subsystem.
compartment	is the number of the compartment for which drawing attributes are to be set or displayed. The compartment must be specified in this form of the LOCAte command. If no further arguments are included, CONSAM will display the drawing attributes for the compartment.
x-coord	is the horizontal coordinate for positioning the compartment.
y-coord	is the vertical coordinate for positioning the compartment.
radius	is the radius of the circle in which the compartment figure is inscribed.
o-angle	is the angle of the arrow representing L(O,compartment).
i-angle	is the angle of the arrow representing IC(compartment).
u-angle	is the angle of the arrow representing UF(compartment).
k-angle	is the angle of the triangle representing K(compartment).

The LOCAte command with the /ALL qualifier is used to set or display the drawing attributes for all compartments, active or inactive, in a subsystem. There are two syntaxes for this form of the LOCA command. Note that the second format allows that an attribute can be set using its first letter as a label:

LOCA/ALL [subsystem] [x-coord [y-coord [radius [o-angle [i-angle [u-angle [k-angle]]]]]]]

LOCA/ALL [subsystem] [X:coord] [Y:coord] [R:radius] [O:angle] [I:angle] [U:angle] [K:angle]

where:

the arguments are identical to those for the "LOCA compartment" command above.

The LOCAte command with the /CURS qualifier may be used to quickly position compartments. In response to this command, CONSAM will enter cursor positioning mode. In this mode, the user moves the terminal's cursor to the desired position in the graphics area and presses the PT (pad terminator) key. To exit cursor mode, type END in response to the compartment number prompt (4020 terminals of version 3 or higher) or press the F1 key (4020 terminals with versions less than 3). The syntax for this form of the LOCA command is:

LOCA/CURS [subsystem]

where:

subsystem — is the subsystem for which compartments are to be positioned. This subsystem will also become the default subsystem. If no subsystem is specified, CONSAM will postion compartments for the default subsystem.

The LOCAte command with /GAP qualifier may be used to set the space between the arrows representing bidirectional transfers (e.g., L(2,1) and L(1,2)):

LOCA/GAP [subsystem] gap

where:

subsystem — is the subsystem for which the gap specification is to be set. This subsystem will also become the default subsystem. If no subsystem is specified, CONSAM will set the gap for the default subsystem.

gap — is an integer specifying in pixels the gap to be used in subsequent DRAWs. A pixel is the smallest representable distance in a graphics display.

The LOCAte command with the /RADI qualifier is used to set or display the lengths of the arrows (as relative radii) representing L(0,J), IC(I) and UF(I). The size of the figures appended to a compartment are proportional to the radius of the compartment. The syntax of the LOCA/RADI command is:

LOCA/RADI [subsystem] [O:o-radius] [I:i-radius] [U:u-radius]

where:

subsystem — is the subsystem for which radius specifications are to be set or displayed. The named subsystem will also become the default subsystem. If no subsystem is specified, CONSAM will set or display radius attributes for the default subsystem.

radius — is the ratio of the arrow length to the radius of the compartment to which it is attached. This number will be used to calculate the length of the appropriate arrow. If no radii are specified, CONSAM will display the current radius specifications for the appropriate subsystem.

o-radius — is the relative radius for the arrow representing L(0,J).

i-radius — is the relative radius for the arrow representing IC(I).

u-radius — is the relative radius for the arrow representing UF(I).

Examples:

LOCA	List the drawing attributes for all active compartments of the default subsystem.
LOCA/ALL G	List drawing attributes for all compartments, active or inactive, of the general model structure.

LOCA INSU 1	List attributes for compartment 1 in model structure INSU. Set the default SUBSystem to INSU.
LOCA 1 .5 .7 .2	Set the coordinates for compartment 1 in the default model structure to (.5,.7) and set the radius to .2.
LOCA/CURS INSU	Use the 4025 cursor to assign X and Y coordinates for compartments in model structure INSU.
LOCA 2 .6 .7 .05 90 0 270 180	Set the attributes for compartment 2 in the default model structure as follows: x-coordinate = .6, y-coordinate = .7, radius = .05, L(0,2) angle = 90, IC(2) angle = 0, UF(2) angle = 270 and K angle = 180.
LOCA 2 K:180 O:90 I:0 U:270	Set the drawing attributes for compartment 2 in the default model structure as follows: K angle = 180, L(0,2) angle = 90, IC(2) angle = 0 and UF(2) angle = 270.
LOCA/ALL O:90	Set the OUT (L(0,J) angle) for all compartments in the default model structure to 90 degrees.
LOCA/ALL O:-10	Remove all the OUTs from this model. (Negative angles cause the compartment's modifier to be suppressed).
LOCA/GAP 10	Set the GAP between bidirectional transfers to 10 pixels.
LOCA/RADI O:2. I:1.5	Set the size of the OUT (L(0,J) to twice the radius and the size of the IC(I) arrow to 1.5 times the radius for all compartments of the default subsystem.

* * * * * REROUTING COMPARTMENTAL CONNECTIONS * * * * *

You may wish to divert an interconnection from a straight path between compartments. This is accomplished by REROuting the path through a user defined point. In addition, this command allows you to define figures in your model by connecting user defined points. The REROute command is also used to display REROute information and to delete REROutes. The syntax for the REROute command is:

RERO[/DELE] [subsystem] [compartment [compartment[...]] compartment]

where:

/DELE	requests that the specified REROute be deleted.
subsystem	specifies the subsystem for which REROute(s) are to be displayed, defined or deleted. This subsystem will become the default subsystem. If no subsystem is specified, CONSAM will act on the default subsystem.
compartment	is a positive or negative compartment number. If no compartment numbers are specified, CONSAM will display all REROutes in the appropriate subsystem. At least two compartments (the begining and end of the REROuted connection) must be specified to define or delete a REROute.

Examples:

RERO INSU	List the rerouting entries in model structure INSU.
RERO 1 -1 2	Reroute the connection from compartment 1 to compartment 2 (the arrow representing L(1,2)) via user defined point -1. If -1 has not been defined, do not connect 1 and 2.
RERO -2 -3 -4 -2	Draw a triangle between points -2, -3, and -4.
RERO/DELE 1 2	Delete the rerouting entry(s) connecting compartments 1 and 2.

* * * * * SUBSYSTEMS * * * * *

You may wish to define several different model structures or use a subset of the full model. Either may be accomplished using the SUBSystem command.

The syntax for defining or changing a SUBSystem is:

```
SUBS name={[subsystem],comp}[{*,/}{subsystem,comp}][...]
```

where:

name	is the name of the subsystem to be defined or changed. A subsystem name may consist of from 1 to to 4 alphanumeric characters and must start with a letter. The subsystem name G refers to the general model structure. The subsystem defined will become the default subsystem.
subsystem	is the name of a previously defined subsystem or the letter G. If the first subsystem after the equal sign (=) is omitted, the default subsystem will be used.
comp	is a positive or negative compartment number (an integer between -25 and 25 inclusive, except 0).
*	indicates that the subsystem or compartment following the the * is to be included in the subsystem.
/	indicates that the compartment following the / or all compartments in the subsystem following the / are to be removed from the subsystem.
[...]	indicates that more than one * or / operation with an accompanying subsystem name or compartment number may be used. CONSAM processes the list from left to right, including and excluding compartments and subsystems as requested. A comma may be used in the definition list and has the same effect as a *.

Examples:

SUBS LIPO=INSU*1/-2	Define LIPO as a model structure containing the model structure INSU plus compartment 1 minus the user defined point -2.

SUBS G=G/-2	Remove the user defined point -2 from the general model.
SUBS INSU=LIPO*GLUC	Define INSU as a model structure containing the contents of model structures LIPO and GLUC.
SUBS AA=/BB	Define AA as a model structure containing the default model structure with the compartments in model structure BB removed.
SUBS INSU=1,2,3	Define the model structure INSU containing compartments 1, 2, and 3.

The SUBSystem command may also be used to list the names of defined subsystems or to list the compartments which are active in a subsystem. The syntax for displaying such subsystem information is:

SUBS [subsystem]

where:

subsystem — is the name of the subsystem for which a list of the active compartments is desired. This subsystem will also become the default subsystem. If the subsystem name is omitted, CONSAM will display a list of all defined subsystems.

Examples:

SUBS	Display the names of the available model structures.
SUBS INSU	Display the compartments in the model structure INSU.

The SUBSystem command may also be used to delete defined SUBSystems. The general subsystem G may not be deleted. If the default subsystem is deleted, G will become the default subsystem. The syntax for deleting a subsystem is:

SUBS/DELE subsystem

where:

subsystem — is the name of the user defined subsystem to be deleted. This argument is required.

Example:

```
SUBS/DELE INSU          Delete subsystem INSU from the directory.
```

****

MATRIX CALCULATIONS

****

CONSAM contains ten matrix "spaces", each consisting of a 25x25 matrix and a 1x25 vector. Some of these have specific interpretations. Following is a list of these matrix and vector spaces:

Matrix name	Vector name	Interpretation or use
LI	LIV	Inverse of L matrix (L-inverse)
LM	LMV	L matrix
EIG	EGV	Eigenvectors and eigenvalues
MTX	VCT	Scratch user space
SIM	SMV	Similarity transformations
INV	NVC	Inverse matrix
LS	LSV	
PS	PSV	
LME	LEV	
LMI	LVI	

The CALCulate command may be used to calculate a matrix which has a specific interpretation:

CALC LM — Perform a solve and saves the L matrix for TC(0) in LM. Requires KOMN. Time-variant component of fn-dependent parameters is neglected.

CALC LI — Perform a solve and saves the LM-Inverse matrix for TC(0) in LI. Requires KOMN. Time-variant component of fn-dependent parameters is neglected. The element LI(I,J) is the mean residence time in compartment I for particles which enter the system via compartment J. LI(I,J) is also the mass in compartment I generated by a unit input rate into compartment J. For a unit tracer injection into compartment J, LI(I,J) is the area under the tracer curve observed in compartment I.

CALC EIG matrix — Calculate eigenvectors and eigenvalues for the specified matrix and saves them in EIG and EGV respectively.

CALC INV matrix — Calculate the inverse of the specified matrix and save it in INV.

CALC SIM p-matrix {matrix,vector}

Perform a similarity transformation and save the result in SIM:

SIM=INV(P)*L*P

where P is p-matrix and INV(P) is the inverse of p-matrix. If a matrix is specified as the second argument, L is the specified matrix. If a vector is specified as the second argument, L is the diagonal matrix with the elements of the specified vector as its diagonal elements: L(I,I)=vector(I).

where:

matrix	is a matrix name (e.g., MTX)
vector	is a vector name (e.g., VCT)

The COPY command may be used to transfer the contents of one matrix space to another. The syntax for thge COPY command is:

COPY matrix-1 matrix-2 {V,M,[B]}

where:

matrix-1	is the matrix name for the space to be copied from.
matrix-2	is the matrix name for the space to be copied to.
M	specifies that only the matrix is to be copied.
V	specifies that only the vector is to be copied.
B	specifies that both the matrix and vector are to be copied. This is the default if the M, V or B argument is omitted.

Examples:

COPY LM MTX	Copy the contents of LM into MTX and the contents of LMV into VCT.
COPY EIG MTX V	Copy the contents of EGV into VCT.

The MATRices command gives a status and size report for all the matrix spaces. The syntax for the MATRices command is

MATR Display status and size information on matrix spaces.

The OUTPut command may be used to write matrices and/or vectors to a disk file. The INPUt command may be used to read matrices and/or vectors from a disk file. The units for these operations are 38 for OUTPut and 37 for INPUt. Matrix and vector contents are written in H MAT format. See the SAAM Manual for details. The syntaxes for the OUTPut and INPUt commands are:

OUTP {matrix,vector,[ALL]}

Write, in SAAM format, the specified matrix(s) or vector(s) to logical unit 38.

INPU {matrix,vector,[ALL]}

Read the specified matrix(s) or vector(s) from logical unit 37

where:

matrix — is one of the matrix names:
LI LM EIG MTX SIM INV LS PS LME LMI

vector — is one of the vector names:
LIV LMV EGV VCT SMV NVC LSV PSV LEV LVI

ALL — specifies that all non-NULL matrices and vectors are to be OUTPut or all matrices and vectors found on unit 37 are to INPUt. ALL is the default if no matrix or vector is specified.

Matrices and vectors can be accessed via the (GET) command:

matrix({I,index},{J,index})

vector({I,index})

where:

matrix — is one of the matrix names:
LI LM EIG MTX SIM INV LS PS LME LMI

vector — is one of the vector names:
LIV LMV EGV VCT SMV NVC LSV PSV LEV LVI

index — is an integer between 1 and 25.

Examples:

LM(2,J)	Display the 2nd row of L matrix LM.
MTX(I,3)	Display the 3rd column of the scratch matrix MTX.
EGV(I)	Display all elements of the eigenvalue vector EGV.

Elements of matrices and vectors may be assigned values using the (SET) command:

matrix(index,index)=value

vector(index)=value

where:

matrix — is one of the matrix names:
LI LM EIG MTX SIM INV LS PS LME LMI

vector — is one of the vector names:
LIV LMV EGV VCT SMV NVC LSV PSV LEV LVI

index — is an integer between 1 and 25.

value — is a number to be assigned to the specified matrix or vector element.

Examples:

```
MTX(1,2)=.5

VCT(4)=0.
```

After a CALC LI has been performed, elements of the residence time (TR) matrix and the fractional catabolic rate (FCR) matrix are available. The TR matrix is the row-wise normalized L-inverse matrix. The value of TR(I,J) is calculated as TR(I,J)=LI(I,J)/LI(I,I) and corresponds to the fraction of particles in compartment J which will reach compartment I. The value of FCR(I,J) is calculated as 1/LI(I,J). The TR and FCR matrices do not have corresponding matrix spaces but are calculated from the contents of the LI matrix space. Elements of the TR and FCR matrices may be examined using the (GET) command:

TR({compartment,I},{compartment,J})

Display element(s) of the normalized LI matrix.

FCR({compartment,I},{compartment,J})

Display element(s) of the FCR matrix.

where:

compartment is an integer between 1 and 25.

****

CONSAM CALCULATOR
by S. JOFFE

****

The EVALuate command activates the CONSAM calculator and puts CONSAM in calculator mode. The CONSAM calculator is based on the HP 25. In calculator mode, the prompt is: C> . After every command, the calculator displays the value in the X register. Most CONSAM commands are executable in calculator mode.

The calculator commands are:

EVAL	Enter calculator mode.
EXIT	Leave calculator mode.
ENT	Enter: Shift the Z register value into register T, Y entry into Z and X into Y. The X register is unchanged.
STO {1-10,parameter}	Store the X register value in the specified storage register or as the value of the specified parameter. There are a total of 10 storage registers.
RCL {1-10}	Recall the value from the indicated storage register and enter it in the X register.
R	Roll down the X, Y, Z and T registers.
FIX [places-after-decimal]	Display numbers in fixed (F) format with the specified number of places after the decimal point. If no argument is entered, the calculator will display numbers with 4 places after the decimal.
SCI [places-after-decimal [places-before-decimal]]	Display numbers in scientific notation (E format) with the specified number of figures before and after the decimal point. If no arguments are entered, the default is 9 places after the decimal and 1 place before.
DEG	Set calculator in degree mode.

RAD	Set calculator in radian mode.
GRD	Set calculator in gradient mode.

Examples:

STO 1	Store the X register value in register 1.
STO L(1,2)	Give L(1,2) the value in the X register.
RCL 1	Recall the value from register 1 and store it in the X register.
FIX 5	Display numbers in fixed (F) format with 5 figures after the decimal point: 3.14159
SCI 5 3	Display numbers in scientific notation with 5 figures after the decimal point and 3 figures before the decimal point: 314.15927E-02
SCI 5	Display numbers in scientific notation with 5 figures after the decimal point and 1 figure before the decimal point: 3.14159E+00

Parameter values and numbers may be brought into the calculator by entering the parameter name or the number. The special name PI may be used to bring the value of pi (3.1415...) into the calculator. Any of these "commands" causes the Z register value to be shifted into the T register, the Y value into the Z register, the X value into the Y register and the specified value into the X register:

parameter	Store the value of the specified parameter in the X register roll up the old X, Y and Z values.
{number,PI}	Store the specified value in the X register and roll up the old X, Y and Z values.

Examples:

L(0,1)	Store the value of L(0,1) in the X register.
W(1)	Store the weight of datum 1 in the X register.

9.322	Enter the value 9.322 in the X register.
PI	Store the value 3.141... in the X register.

The following operations can be performed on register contents. Single argument operations act on the value in the X register, store the result in the X register and leave the Y, Z and T registers intact. Two argument operations (with the exceptions >R and >P) act on the values in the X and Y registers, store the result in the X register and roll down the Z value into the Y register and the T value into the Z register.

*	Y times X
+	Y plus X
-	Y minus X
/	Y divided by X
Y^X	Y to the X power
X^Y	X to the Y power
E^X	e to the X power
X^2	X squared
SQRT	square root of X
ILOG	10 to the X power (InverseLOG)
LN	natural log of X
LOG	log base 10 of X
INV	1/X
CHS	-X (change sign)
ABS	absolute value of X
X><Y	switch values in X and Y registers
SIN	sine of X
COS	cosine of X
TAN	tangent of X
ASIN	arcsine of X
ACOS	arccosine of X
ATAN	arctangent of X
INT	integer portion of X (truncation)
FRAC	fractional portion of X (X-INT(X))
LAST	recover X before the previous operation
CLX	clear X register (X=0)
CLR	clear the stack (X=0, Y=0, Z=0, T=0)
REG	clear storage registers
%	X percent of Y (Y, Z and T remain unchanged)
>R	convert to rectangular coordinates. The value in the Y register is taken as the angle and the value in X as the radius. X returns with the x-coordinate and Y with the y-coordinate. Z and T are unchanged.

>P	convert to polar coordinates. The value in the Y register is taken as the y-coordinate and the value in the X register as the x-coordinate. X returns with the radius and Y with the angle. Z and T are unchanged.
SUM+	summation + see HP manual for details
SUM-	summation - see HP manual for details
AVG	average of numbers stored by SUM+ and SUM-
STD	standard deviation of numbers stored by SUM+ and SUM-

Following is an alphabetical listing of all calculator commands:

%	x percent of y
*	multiply
+	add
-	subtract
/	divide
>P	convert to polar coordinates
>R	convert to rectangular coordinates
ABS	absolute value of x
ACOS	arccosine of x
ASIN	arcsine of x
ATAN	arctangent of x
AVG	average of numbers stored by SUM+ and SUM-
CHS	change sign (x=-x)
CLR	clear the x,y,z and t registers
CLX	clear x register
COS	cosine of x
DEG	set degree mode
E^X	e to the x power
ENT	shift X register into Y, Y into Z, Z into T
EXIT	leave calculator mode
FIX [places-after-decimal]	display values in F format
FRAC	fractional portion of x
GRD	set gradient mode
ILOG	10 to the x power
INT	integer portion of x
INV	1/x
LAST	x before previous operation
LN	natural log of x
LOG	log base 10 of x
PI	value of pi
R	roll down x,y,z,t registers
RAD	set radian mode
RCL {1-10}	recall value from the indicated register

REG	clear all storage registers
SCI [places-after-decimal[places-before-decimal]]	display values in scientific notation
SIN	sine of x
SQRT	square root of x
STD	standard deviation of numbers stored with SUM+ and SUM-
STO {1-10,parameter}	store x in register or as parameter value
SUM+	summation + see HP manual for details
SUM-	summation - see HP manual for details
TAN	tangent of x
X^2	x squared
X^Y	x to the y power
X><Y	switch x and y registers
Y^X	y to the x power
number	store indicated number in x register
parameter	store value of indicated parameter in x register

APPENDICES

****

LOGICAL UNITS USED IN CONSAM

****

During execution, CONSAM writes and reads a number of disk files which are assigned logical unit (device) numbers of 2 - 40. Some of these disk files will remain at the end of a CONSAM session. A few of them contain essential information and should not be deleted arbitrarily.

The table below summarizes logical unit used by CONSAM, their purpose and hence the necessity for retaining them.

LOGICAL UNIT	PURPOSE	COMMENTS
2	Edit buffer	
3	Source input file	Save to run problem again later.
4	Batch style log file	Useful for finding errors.
5	Input (Terminal)	
6	Output (Terminal)	
8	Scratch image of deck	
9	Input deck update	
10	Scratch file	
11	Time interrupt data	May be needed for READ. Retain with disk copy of KOMN.
12	Partials matrix	
13	Time interrupt data	May be needed for READ. Retain with disk copy of KOMN.
14	Scratch file	
15	Critical points data	
16	Plot name directory	Save for access to defined names.
17	Plot point assembler	
22-29	Disk copies of KOMN	Retain to save WRITten KOMN.
30-35	Solution storage utility	Retain to save KEEPed solutions.
37	Matrix input	
38	Matrix output	Retain to save OUTPut matrices.
40	Draw subsystem directory	Save for access to defined subsystems.

****

SWITCH SUMMARY

****

Following is a summary of CONSAM switches, their default settings, the devices for which the switch option is available and a brief reminder of the effect of the switch.

		device:	4	4	1			
			0	0	2		2	
			2	1	0	3	3	
	default		5	0	0	3	0	ON: / OFF:
AUTO	ON		X	X	X	X	X	automatic plot scaling / plots not rescaled
CONN	OFF		X	-	-	-	-	lines between QO's / QO's not connected
FLOW	OFF		X	X	X	X	X	values outside graph indicated by o or u / values outside graph not indicated
FRAM	OFF		X	X	-	-	-	graph enclosed in box / coordinate axes only
HCOP	OFF		X	-	-	-	-	graph expanded for copies / graph not expanded
LABE	ON		X	X	+	+	X	single plot mode: ticks and numbers drawn / no ticks or numbers; multiple plot mode: axes drawn / axes not drawn
LINL	ON		X	X	X	X	X	linear region at bottom of semilog plots / no linear region on semilog plots
LOG	ON		X	X	X	X	X	semilog plots / linear plots
MULT	OFF		X	X	-	-	X	multiple plot mode / single plot mode
PART	OFF		X	X	X	X	X	partials calculated / no calculation
STOR	OFF		X	X	X	X	X	automatic solution saving / solutions not saved
STRE	*		X	X	X	X	X	T-axis distorted to display all points / no distortion, points may overlap

* The default for the STREtch switch is OFF for DEVIces 4025, 4010 and 230, ON for DEVIces 33 and 1200

Key:
- X Option is available with this device
- \+ Option not available: switch is effectively always ON
- \- Option not available: switch is effectively always OFF

CONSAM COMMAND SUMMARY

Following is a list of CONSAM commands showing the syntax for each command. Expressions used are defined at the end of the list. A command marked with a single asterisk (*) may prompt the user for more information. A command marked with a double asterisk (**) may not be used in calculator mode. Calculator commands are not listed in this summary.

PROGRAM EXECUTION

DECK	*	process source input file to create KOMN
MODE[={1-5,7,8,10,11,21-23}]		set/display model code
SOLV		compute solution for model in KOMN
DEPE		solve for dependent parameter values
ITER		perform 2 iterations
PART		compute partials
RESI		calculate correction vector
OPTI		calculate new adjustable parameters values
parameter=value		assign value parameter
PROJ		use pseudo-partial estimates
EXCL {ALL,parameter[,...]}		exclude parameters from adjustment
INCL {ALL,parameter[,...]}		include parameters for adjustment
UNWE {name,ALL} [time[time]]		assign zero weights to data
REWE {name,ALL} [time[time]]		restore original data weights
MAX parameter value		set adjustable parameter upper limit
MIN parameter value		set adjustable parameter lower limit
STOP	*	save edit buffer and leave CONSAM

EDITING

CURR		display current line number
P [{DECK,CURR,NEXT,LAST,line[line],END,EQUA}]		display appropriate line(s)
P STRI	*	search for and display character string
integer	**	display line relative to current line
I [line]	*	insert text
A {C,D,I,R} [column]	*	alter current line
CHAN	*	replace one character string with another
C	*	repeat last CHANge command

```
DELE [line[ line]]                      delete line(s) in edit buffer
TABS  [{column [...],DATA,ICC,PAR,PCC}]
                                        set/display tab stops
FIX  {g-parameter,ALL}  [value]   **    remove edit buffer min and max values
FREE {g-parameter,ALL} [value[ value]]
                                        write edit buffer min and max values
RESC                                    copy source input file into edit buffer
RETA                                    copy edit buffer into source input file
UPDA                                    replace edit buffer parameter values
```

DISPLAY OF RESULTS

```
PRIN                                    print data of last PRIN, PLOT or ERAS
PRIN data-type(name[,...]) [solution[,...]]
                                        print data
PRIN {P,S}(name[,...]) parameter[,...]
                                        print partials or sensitivities
PLOT                                    plot data of last PRIN, PLOT or ERAS
PLOT data-type(name[*factor][#lint][,...]) [solution[,...]]
                                        plot data
PLOT data-type(name[*factor][,...]) [solution[#lint][,...]]
                                        plot data
PLOT {P,S}(name[*factor][#lint][,...])  parameter[,...]]
                                        plot partials or sensitivities
PLOT {P,S}(name[*factor][,...])  parameter[#lint][,...]]
                                        plot partials or sensitivities
g-parameter                             display parameter information
matrix({I,1-25},{J,1-25})               display matrix element(s)
vector({I,1-25})                        display vector element(s)
M                                       display number of data
M2                                      display # of data + # of stat. constraints
Q(I)                                    list components containing data
SS[({I,component})]                     display sum(s) of squares
SD({adj-index,I})                       display parameter standard deviations
FSD({adj-index,I})                      display parameter fractional sd's
COR({adj-index,I},{adj-index,J})        display adjustable parameters correlation
RES({adj-index,I})                      display correction vector component(s)
G(index[,{time,T}][,C:component])       display data values or evaluate G function
function(compartment[,{time,T}][,C:component])
                                        display datum value or evaluate a function
T[=value]                               display/set current T
TC[=integer]                            display/set current TC block
TR({compartment,I},{compartment,J})     display normalized LI(I,J) element(s)
FCR({compartment,I},{compartment,J})    display elements from the FCR matrix
```

```
STAT                                    display all switch settings
TIME                                    display cumulative CPU time
TITL                                    display problem title
KOMN  [D]                               display information on KOMN
ADJU [{DT,K,L,P,S,UF}]                  list adjustable parameters
LIST {DEVI,MODEL}                       list options
INFO                                    perform and display informational analysis
```

PLOTTING

```
PLOT                                    plot data of last PRIN, PLOT or ERAS
PLOT data-type(name[*factor][#lint][,...]) [solution[,...]]
                                        plot data
PLOT data-type(name[*factor][,...]) [solution[#lint][,...]]
                                        plot data
PLOT {P,S}(name[*factor][#lint][,...])  parameter[,...]]
                                        plot partials or sensitivities
PLOT {P,S}(name[*factor][,...])  parameter[#lint][,...]]
                                        plot partials or sensitivities
DEVI={33,230,1200,4010,4025}         * set output device type
TSCA name [{value[ value],C}]           clear/set/display T axis scaling
YSCA name [{value[ value],C}]           clear/set/display Y axis scaling
TRAN name [{value[ value],C}]           clear/set/display data T range for display
YRAN name [{value[ value],C}]           clear/set/display data Y range for display
CYCL  name  [{C,1,2,3,4}]               clear/set/display cycle number for plots
SCAL name [{t-factor[ y-factor],C}]     clear/set/display plot scale factors
AXIS name [{t-length y-length,C}]       clear/set/display axis lengths
OFFS name [{C,t-coord y-coord}]         clear/set/display plot origin offset
LINT [lint]                             set/display plotting line type(s)
ERAS                                    erase data of last PRIN, PLOT or ERAS
ERAS data-type(name[*factor][#lint][,...]) [solution[,...]]
                                        erase graphic data
ERAS data-type(name[*factor][,...]) [solution[#lint][,...]]
                                        erase graphic data
ERAS {P,S}(name[*factor][#lint][,...])  parameter[,...]
                                        erase partials or sensitivities
ERAS {P,S}(name[*factor][,...])  parameter[#lint][,...]
                                        erase partials or sensitivities
```

DRAWING

```
DRAW[/AUTO][/NORE][/MINU][/GRID][/ERAS][/NOLA]  [subsystem]
                                        draw the model or subsystem thereof
LOCA [subsystem]                        display drawing parameters
LOCA [subsystem] compartment [X:coord] [Y:coord] [R:radius] [O:angle]
                             [I:angle] [U:angle] [K:angle]
                                        set/display drawing parameter(s)
LOCA [subsystem] compartment [x-coord [y-coord [radius [o-angle [i-angle
                             [u-angle [k-angle]]]]]]]
                                        set/display drawing parameter(s)
LOCA/ALL [subsystem] [X:coord] [Y:coord] [R:radius] [O:angle] [I:angle]
                             [U:angle] [K:angle]
                                        set/display drawing parameter(s)
LOCA/ALL [subsystem] [x-coord [y-coord [radius [o-angle [i-angle [u-angle
                             [k-angle]]]]]]]
                                        set/display drawing parameter(s)
LOCA/CURS [subsystem]                 * position compartments in cursor mode
LOCA/GAP [subsystem] integer            set the gap (in pixels) between arrows
LOCA/RADI [subsystem] [O:radius] [I:radius] [U:radius]
                                        set/display relative radius
RERO[/DELE] [subsystem] [compartment compartment [...]]
                                        define/delete/display reroute(s)
SUBS  [subsystem]                       display subsystem information
SUBS  subsystem=compartment[,...]
                                        define a subsystem
SUBS  subsystem=subsystem[{*{subsystem,compartment},/compartment}][,...]
                                        define a subsystem
SUBS/DELE  user-defined-subsystem       delete a subsystem
```

MATRIX CALCULATIONS

```
MATR                                    display matrix information
matrix({I,1-25},{J,1-25})               display matrix element(s)
matrix({1-25},{1-25})=value             set matrix element value
vector({I,1-25})                        display vector element(s)
vector({1-25})=value                    set vector element value
CALC EIG matrix                         compute eigenvectors and eigenvalues
CALC INV matrix                         compute inverse matrix
CALC {LI,LM}                            compute matrix
CALC SIM matrix {matrix,vector}         perform similarity transformation
COPY matrix matrix {M,V,[B]}            copy matrix and/or vector
```

```
INPU {matrix,vector,[ALL]}            read matrix(es) and/or vector(s)
OUTP  {matrix,vector,[ALL]}           output matrix(es) and/or vector(s)
TR({compartment,I},{compartment,J})   display normalized LI(I,J) element(s)
FCR({compartment,I},{compartment,J}) display elements from the FCR matrix
```

MODEL AND SOLUTION SAVING UTILITIES

```
WRIT [{22-29}]                        save KOMN on disk
READ [{22-29}]                        read KOMN from disk
NEW  [{30-35}]                        create new REFIL
OLD  [{30-35}]                        open old REFIL
KEEP                                  save solution in REFIL
REST  solution                        restore solution from REFIL
RELI  solution                        display REFIL parameter values
RECO  [D]                             display information on REFIL records
COUN  record                          reset the number of stored solutions
MOVE  record>record                   move solution record
UNIT                                  diplay REFIL and saved KOMN unit numbers
BPLO                                  send batch style plots to unit 4
BPRI                                  send batch style output to unit 4
EX DECK                               execute edit buffer as CONSAM commands
```

MISCELLANEOUS COMMANDS

```
EVAL                                  enter calculator mode
SWIT  switch  {[ON],O[FF]}            set switch
NAME  [name[/F]]                   ** display name information
NAME  user-defined-name=C:component[<[TC(integer)][,][category]>[,...]
                                   ** define name
NAME  user-defined-name/DELE       ** delete name
RAND[/{NORM,EXP,RECT}][/{FSD,SD,RQO}[=value]][/SEED=value][/BIAS=value]
                     Q(component)
                                      generate QO values with random errors
```

EXPRESSIONS:

adj-index	adjustable parameter index: an integer between 1 and the number of adjustable parameters inclusive
angle	angle in degrees
category	an allowed SAAM data category, see SAAM manual
column	edit buffer column number: an integer
compartment	compartment number: {1-25}
component	data component number: {0-25}
coord	graphics coordinate as fraction of graphics area: a real number
data-type	type of data to be displayed: {Q,QO,QC,FSE,R,RES,TH,W}
factor	multiplication factor: a number
function	compartment related function type: {F,FF,QO,UF}
g-parameter	global parameter: { parameter, DN(I), DT(I), G(I), IC(I), K(I), L(I,J), L(compartment,J), L(I,compartment), M(I), P(I), R(I,J), R(compartment,J), R(I,compartment), S(I,J), S(compartment,J), S(I,compartment), U(I), UF(I)}
index	G, K or P index: {1-99}
length	axis length as fraction of graphics area: a real number
line	edit buffer line number: a positive integer
lint	line type for plotting: {1,2,3,4,5}
matrix	matrix name: {LI,LM,EIG,MTX,SIM,INV,LS,PS,LME,LMI}
name	{G,0-25,user-defined-name}
parameter	specific parameter in the model: { DN(compartment), DT(compartment), G(index), IC(compartment), K(index), L({0,compartment},compartment), M(compartment), P(index), R({0,compartment},compartment), S(compartment,compartment), U(compartment), UF(compartment) }
radius	radius relative to compartment radius: a real number
record	REFIL record number: an integer between 1 and the number of records stored in REFIL
solution	REFIL solution reference: {integer,C}
subsystem	{G,user-defined-subsystem}
switch	{ AUTO, CONN, FLOW, FRAM, HCOP, LABE, LINL, LOG, MULT, PART, STOR, STRE }
time	value of independent variable: a number
user-defined-name	alphanumeric string up to 4 characters long, the first character must be alphabetic
user-defined-subsystem	alphanumeric string up to 4 characters long, the first character must be alphabetic
value	a number, appropriate values depend on usage
vector	a vector name: {LIV,LMV,EGV,VCT,SMV,NVC,LSV,PSV,LEV,LVI}

****

ALPHABETICAL LIST OF COMMANDS

****

Following is an alphabetical listing of CONSAM commands showing the syntax for each command. Expressions used are defined at the end of the list. A command marked with a single asterisk (*) may prompt the user for more information. A command marked with a double asterisk (**) may not be used in calculator mode. Calculator commands are not listed in this summary.

```
<CR>                                        (carriage return), display next line
g-parameter                                 display parameter information
function(compartment[,{time,T}][,C:component])
                                            display datum value or evaluate a function
integer                                  ** display line relative to current line
matrix({I,1-25},{J,1-25})                   display matrix element(s)
matrix({1-25},{1-25})=value                 set matrix element value
parameter[=value]                           display/set parameter value
vector({I,1-25})                            display vector element(s)
vector({1-25})=value                        set vector element value
A  {C,D,I,R}  [column]                   *  alter current line
ADJU [{DT,K,L,P,S,UF}]                      list adjustable parameters
AXIS name [{t-length y-length,C}]           clear/set/display axis lengths
BPLO                                        send batch style plots to unit 4
BPRI                                        send batch style output to unit 4
C                                        *  repeat last CHANge command
CALC EIG matrix                             compute eigenvectors and eigenvalues
CALC INV matrix                             compute inverse matrix
CALC {LI,LM}                                compute matrix
CALC SIM matrix {matrix,vector}             perform similarity transformation
CHAN                                     *  replace one character string with another
COPY matrix matrix {M,V,[B]}                copy matrix and/or vector
COR({adj-index,I},{adj-index,J})            display adjustable parameters correlation
COUN  record                                reset the number of stored solutions
CURR                                        display current line number
CYCL  name  [{C,1,2,3,4}]                   clear/set/display cycle number for plots
DECK                                     *  process source input file to create KOMN
DELE [line[ line]]                          delete line(s) in edit buffer
DEPE                                        solve for dependent parameter values
DEVI={33,230,1200,4010,4025}             *  set output device type
DRAW[/AUTO][/NORE][/MINU][/GRID][/ERAS][/NOLA]  [subsystem]
                                            draw the model or subsystem thereof
ERAS                                        erase data of last PRIN, PLOT or ERAS
ERAS data-type(name[*factor][#lint][,...]) [solution[,...]]
                                            erase graphic data
ERAS data-type(name[*factor][,...]) [solution[#lint][,...]]
                                            erase graphic data
```

```
ERAS {P,S}(name[*factor][#lint][,...])  parameter[,...]
                                        erase partials or sensitivities
ERAS {P,S}(name[*factor][,...])  parameter[#lint][,...]
                                        erase partials or sensitivities
EVAL                                    enter calculator mode
EX DECK                                 execute edit buffer as CONSAM commands
EXCL  {ALL,parameter[,...]}             exclude parameters from adjustment
FCR({compartment,I},{compartment,J}) display elements from the FCR matrix
FIX  {g-parameter,ALL}  [value]   ** remove edit buffer min and max values
FREE {g-parameter,ALL} [value[ value]]
                                        write edit buffer min and max values
FSD({adj-index,I})                      display parameter fractional sd's
G(index[,{time,T}][,C:component])    display data values or evaluate G function
(GET) {F,FF,QO,UF}(compartment[,{time,T}][,C:component])
                                        display data values or evaluate function
(GET) g-parameter                       display parameter information
(GET) matrix({I,1-25},{J,1-25})         display matrix element(s)
(GET) vector({I,1-25})                  display vector element(s)
I  [line]                             * insert text
INCL  {ALL,parameter[,...]}             include parameters for adjustment
INFO                                    perform and display informational analysis
INPU {matrix,vector,[ALL]}              read matrix(es) and/or vector(s)
ITER                                    perform 2 iterations
KEEP                                    save solution in REFIL
KOMN  [D]                               display information on KOMN
LINT [lint]                             set/display plotting line type(s)
LIST {DEVI,MODEL}                       list options
LOCA [subsystem]                        display drawing parameters
LOCA [subsystem] compartment [X:coord] [Y:coord] [R:radius] [O:angle]
                             [I:angle] [U:angle] [K:angle]
                                        set/display drawing parameter(s)
LOCA [subsystem] compartment [x-coord [y-coord [radius [o-angle [i-angle
                             [u-angle [k-angle]]]]]]]
                                        set/display drawing parameter(s)
LOCA/ALL [subsystem] [X:coord] [Y:coord] [R:radius] [O:angle] [I:angle]
                             [U:angle] [K:angle]
                                        set/display drawing parameter(s)
LOCA/ALL [subsystem] [x-coord [y-coord [radius [o-angle [i-angle [u-angle
                             [k-angle]]]]]]]
                                        set/display drawing parameter(s)
LOCA/CURS [subsystem]                 * position compartments in cursor mode
LOCA/GAP [subsystem] integer            set the gap (in pixels) between arrows
LOCA/RADI [subsystem] [O:radius] [I:radius] [U:radius]
                                        set/display relative radius
M                                       display number of data
M2                                      display # of data + # of stat. constraints
```

```
MATR                                        display matrix information
MAX parameter value                         set adjustable parameter upper limit
MIN parameter value                         set adjustable parameter lower limit
MODE[={1-5,7,8,10,11,21-23}]                set/display model code
MOVE  record>record                         move solution record
NAME  [name[/F]]                         ** display name information
NAME  user-defined-name=C:component[<[TC(integer)][,][category]>[,...]
                                         ** define name
NAME  user-defined-name/DELE             ** delete name
NEW  [{30-35}]                              create new REFIL
OFFS name [{C,t-coord y-coord}]             clear/set/display plot origin offset
OLD  [{30-35}]                              open old REFIL
OPTI                                        calculate new adjustable parameters values
OUTP  {matrix,vector,[ALL]}                 output matrix(es) and/or vector(s)
P  [{DECK,CURR,NEXT,LAST,line[ line],END,EQUA}]
                                            display appropriate line(s)
P STRI                                    * search for and display character string
PART                                        compute partials
PLOT                                        plot data of last PRIN, PLOT or ERAS
PLOT data-type(name[*factor][#lint][,...]) [solution[,...]]
                                            plot data
PLOT data-type(name[*factor][,...]) [solution[#lint][,...]]
                                            plot data
PLOT {P,S}(name[*factor][#lint][,...])  parameter[,...]]
                                            plot partials or sensitivities
PLOT {P,S}(name[*factor][,...])  parameter[#lint][,...]]
                                            plot partials or sensitivities
PRIN                                        print data of last PRIN, PLOT or ERAS
PRIN data-type(name[,...]) [solution[,...]]
                                            print data
PRIN {P,S}(name[,...]) parameter[,...]
                                            print partials or sensitivities
PROJ                                        use pseudo-partial estimates
Q(I)                                        list components containing data
RAND[/{NORM,EXP,RECT}][/{FSD,SD,RQO}[=value]][/SEED=value][/BIAS=value]
                     Q(component)
                                            generate QO values with random errors
READ  [{22-29}]                             read KOMN from disk
RECO  [D]                                   display information on REFIL records
RELI  solution                              display REFIL parameter values
RERO[/DELE] [subsystem] [compartment compartment [...]]
                                            define/delete/display reroute(s)
RES({adj-index,I})                          display correction vector component(s)
RESC                                        copy source input file into edit buffer
RESI                                        calculate correction vector
REST  solution                              restore solution from REFIL
```

```
RETA                                    copy edit buffer into source input file
REWE {name,ALL} [time[ time]]           restore original data weights
SCAL name [{t-factor[ y-factor],C}]     clear/set/display plot scale factors
SD({adj-index,I})                       display parameter standard deviations
(SET) matrix({1-25},{1-25})=value       assign value to matrix element
(SET) {parameter,T,TC}=value            assign value to parameter, T or TC
(SET) vector({1-25})=value              assign value to vector element
SOLV                                    compute solution for model in KOMN
SS[({I,component})]                     display sum(s) of squares
STAT                                    display all switch settings
STOP                                  * save edit buffer and leave CONSAM
SUBS  [subsystem]                       display subsystem information
SUBS  subsystem=compartment[,...]
                                        define a subsystem
SUBS  subsystem=subsystem[{*{subsystem,compartment},/compartment}][,...]
                                        define a subsystem
SUBS/DELE  user-defined-subsystem       delete a subsystem
SWIT  switch  {[ON],O[FF]}              set switch
T[=value]                               display/set current T
TC[=integer]                            display/set current TC block
TABS  [{column [...],DATA,ICC,PAR,PCC}]
                                        set/display tab stops
TIME                                    display cumulative CPU time
TITL                                    display problem title
TR({compartment,I},{compartment,J})     display normalized LI(I,J) element(s)
TRAN name [{value[ value],C}]           clear/set/display data T range for display
TSCA name [{value[ value],C}]           clear/set/display T axis scaling
UNIT                                    diplay REFIL and saved KOMN unit numbers
UNWE {name,ALL} [time[ time]]           assign zero weights to data
UPDA                                    replace edit buffer parameter values
WRIT [{22-29}]                          save KOMN on disk
YRAN name [{value[ value],C}]           clear/set/display data Y range for display
YSCA name [{value[ value],C}]           clear/set/display Y axis scaling
```

EXPRESSIONS:

adj-index	adjustable parameter index: an integer between 1 and the number of adjustable parameters inclusive
angle	angle in degrees
category	an allowed SAAM data category, see SAAM manual
column	edit buffer column number: an integer
compartment	compartment number: {1-25}
component	data component number: {0-25}
coord	graphics coordinate as fraction of graphics area: a real number

data-type	type of data to be displayed: {Q,QO,QC,FSE,R,RES,TH,W}
factor	multiplication factor: a number
function	compartment related function type: {F,FF,QO,UF}
g-parameter	global parameter: { parameter, DN(I), DT(I), G(I), IC(I), K(I), L(I,J), L(compartment,J), L(I,compartment), M(I), P(I), R(I,J), R(compartment,J), R(I,compartment), S(I,J), S(compartment,J), S(I,compartment), U(I), UF(I)}
index	G, K or P index: {1-99}
length	axis length as fraction of graphics area: a real number
line	edit buffer line number: a positive integer
lint	line type for plotting: {1,2,3,4,5}
matrix	matrix name: {LI,LM,EIG,MTX,SIM,INV,LS,PS,LME,LMI}
name	{G,0-25,user-defined-name}
parameter	specific parameter in the model: { DN(compartment), DT(compartment), G(index), IC(compartment), K(index), L({0,compartment},compartment), M(compartment), P(index), R({0,compartment},compartment), S(compartment,compartment), U(compartment), UF(compartment) }
radius	radius relative to compartment radius: a real number
record	REFIL record number: an integer between 1 and the number of records stored in REFIL
solution	REFIL solution reference: {integer,C}
subsystem	{G,user-defined-subsystem}
switch	{ AUTO, CONN, FLOW, FRAM, HCOP, LABE, LINL, LOG, MULT, PART, STOR, STRE }
time	value of independent variable: a number
user-defined-name	alphanumeric string up to 4 characters long, the first character must be alphabetic
user-defined-subsystem	alphanumeric string up to 4 characters long, the first character must be alphabetic
value	a number, appropriate values depend on usage
vector	a vector name: {LIV,LMV,EGV,VCT,SMV,NVC,LSV,PSV,LEV,LVI}

****

TROUBLE SHOOTING

****

Following is a list of common problems which prevent CONSAM from executing:

A. If the problem won't DECK, check that:

1. An A SAAM27 card is present as the first card in the source input file.

2. You are using upper case characters at the terminal.

3. There are no blank lines in the source input file.

If the problem remains, insert a Y card part way down the input file, and try to DECK. Continue to move the Y card down the input file until you locate the problem line. If there is no apparent error, try DELEting and retyping the line or get the file printed out by the printer to check for stray characters.

B. If the problem won't SOLVe, check that:

1. The initial conditions are non-zero.

2. There is a H DAT card in the input file.

If the problem still won't solve, leave CONSAM and read the FOR004.DAT file as this may give more information about the error.

C. General comments

Error messages appear giving the line number containing the error. Note that these numbers are generally 3 lower than the actual line number in the edit buffer, if no card 2,3 or 4 are present in the source input file. Thus the error number refers to the KOMN version, which automatically includes cards 2 to 4.

Secondly, note that if the deck contains Y/Y- cards, these also appear in the edit buffer, but not in KOMN. If the input file contains them, the error messages will not match the line numbers in the edit buffer. If the error is still not obvious after accounting for these factors, we suggest that you submit the job to batch SAAM and scrutinize the output to see where the solution procedure failed.

****
INDEX
****