Guide to Interfacing with ANSYS

ANSYS Release 10.0

002188
August 2005

Guide to Interfacing with ANSYS

ANSYS Release 10.0

ANSYS, Inc.
Southpointe
275 Technology Drive
Canonsburg, PA 15317
ansysinfo@ansys.com
http://www.ansys.com
(T) 724-746-3304
(F) 724-514-9494

Revision History

Number	Release	Date
001382	ANSYS 5.7.1	May 2001
001511	ANSYS 6.1	March 2002
001976	ANSYS 8.1	April 2004
002100	ANSYS 9.0	November 2004
002188	ANSYS 10.0	August 2005

Copyright and Trademark Information

Disclaimer Notice

U.S. GOVERNMENT RIGHTS

Third-Party Software

See the online documentation in the product help files for the complete Legal Notice for ANSYS proprietary software and third-party software. The ANSYS third-party software information is also available via download from the Customer Portal on the ANSYS web page. If you are unable to access the third-party legal notices, please contact ANSYS, Inc.

Published in the U.S.A.

Table of Contents

List of Figures

Preface

Conventions Used in This Guide

This guide uses the following typographic conventions to indicate various types of information:

Convention	Indicates
COMMAND	ANSYS commands. These are shown as uppercase, bold text (for example, **K**, **DDELE**, and so on). In the online documentation, these provide hyperlinks to the appropriate command reference information.
Menu > Item	Menu paths (sometimes referred to as GUI paths). These are shown as bold text with mixed-case, separated by angle brackets ">". An angle bracket indicates a branch to a new menu item.
path/filename.ext	File names, which may or may not include directory paths. These are shown as lower-case, bold text, unless case is significant. Examples are shown with the UNIX directory separator character "/" (slash); if you are using a Microsoft Windows system, use "\" (backslash) as your directory separator character.
ARGUMENT	Arguments for numeric values (such as *VALUE, INC, TIME*) in command syntax. These are shown as upper-case italic text. On some commands, non-numeric convenience labels (for example, ALL and P) can also be entered for these arguments.
Argument	Arguments for alphanumeric values (for example, *Lab* or *Fname*) in command syntax. These are shown in mixed-case, italic letters. The guide also uses italic text for emphasis.
ANSYS Guide Title	The name of an ANSYS manual.
`command,arg1,arg2`	Command input listings, ANSYS output listings, and text that a user enters are shown in fixed-width font.
Note--	Information that supplements the main topic being discussed, such as important tips or guidelines.
Caution:	Actions or situations that could cause problems, unexpected ANSYS behavior, or unexpected results.
Warning	Actions or situations that can shut down ANSYS, damage files, cause loss of data, and so on.

About the Programmer's Guide Set

The ANSYS programmer's guide set provides information about the various programming interfaces available to customers. These manuals assume that you have at least a basic knowledge of programming (a working knowledge of Fortran 90 would be very helpful). The set of four manuals includes:

The APDL Programmer's Guide

This guide was designed for ANSYS users that have some programming skills and wish to tap the power of the ANSYS Parametric Design Language (APDL) to increase the productivity. APDL is a scripting language that is very similar to Fortran 90. The guide describes how to define parameters (variables), how to create macro programs using APDL, how to use APDL for simple user interaction, how to encrypt an APDL macro, and how to debug an APDL macro.

The UIDL Programmer's Guide

The UIDL Programmer's Guide covers the User Interface Design Language (UIDL) including how to modify or construct menus, dialogs and online help from within ANSYS.

Guide To ANSYS User Programmable Features

ANSYS provides a set of Fortran 90 functions and routines that are available to extend or modify the program's capabilities. Using these routines requires relinking the ANSYS program, resulting in a custom version of ANSYS. ANSYS provides an external commands capability which you can use to create shared libraries available to ANSYS (either from ANSI standard C or Fortran 90). you can use this feature to add custom extensions to ANSYS without the need to rebuild the ANSYS executable.

Guide to Interfacing with ANSYS

This guide describes a group of utilities as well as a set Fortran 90 routines that you can use to directly access the ANSYS database. You can also use these capabilities to access data in any of the binary files that ANSYS writes or uses.

Chapter 1: Format of Binary Data Files

1.1. What Are ANSYS Binary Files?

The ANSYS program writes several binary files to store data during an analysis. These files are named **Jobname.ext**, where **Jobname** is the name of the analysis that caused the file to be generated and **.ext** is an extension indicating the type of data in the file. ANSYS-written binary files include the following:

- The following results files, in which the ANSYS program stores the results of solving finite element analysis problems:
  - ***Jobname*.RST** A structural or coupled-field analysis
  - ***Jobname*.RTH** A thermal analysis
  - ***Jobname*.RMG** A magnetic analysis
  - ***Jobname*.RFL** A FLOTRAN analysis
- The ***Jobname*.MODE** file, storing data related to a modal analysis
- The ***Jobname*.RDSP** file, storing data related to a reduced transient analysis.
- The ***Jobname*.RFRQ** file, storing data related to a reduced harmonic analysis
- The ***Jobname*.EMAT** file, storing data related to element matrices
- The ***Jobname*.SUB** file, storing data related to substructure matrices
- The ***Jobname*.TRI** file, storing the triangularized stiffness matrix
- The ***Jobname*.FULL** file, storing the full stiffness-mass matrix

The files listed above cover almost all users' needs, although there are others. For more information, see the *ANSYS Basic Analysis Guide*.

1.1.1. Conventions Used to Describe Binary Files

In the information describing the binary file formats:

- Record ID is the identifier for this record. Not all records will have identifiers; they're indicated only for records whose record pointers are stored in a header.
- Type indicates what kind of information this record stores.
- Number of records indicates how many records of this description are found here.
- Record length indicates the number of items stored in the record.

In some record descriptions, actual variable names used may appear in the record contents area.

1.1.2. The Standard Header for ANSYS Binary Files

Each of the ANSYS program's binary files contains a standard, 100-integer file header that describes the file contents. The header contains the items listed below, always in the order shown:

Item 1	The file number
Item 2	The file format. This item has a value of 1 if the file is small format, -1 if large format.

Item 3	The time, in compact form (i.e., 130619 is 13:06:19)
Item 4	The date, in compact form (i.e., 20041023 is 10/23/2004)
Item 5	The units of measurement used. The value of this item is as follows: • 0 for user-defined units • 1 for SI units • 2 for CSG units • 3 for British units (feet) • 4 for British units (inches)
Item 10	The ANSYS release level in integer form ("*X.X*" in character form)
Item 11	The date of the ANSYS release
Items 12-14	The machine identifier in integer form (three four-character strings)
Items 15-16	The *Jobname* in integer form (two four-character strings)
Items 17-18	The ANSYS product name in integer form (two four-character strings)
Item 19	The ANSYS special version label in integer form (one four-character string)
Items 20-22	The user name in integer form (three four-character strings)
Items 23-25	The machine identifier in integer form (three four-character strings)
Item 26	The system record size
Item 27	The maximum file length
Item 28	The maximum record number
Items 31-38	The *Jobname* (eight four-character strings)
Items 41-60	The main analysis title in integer form (20 four-character strings)
Items 61-80	The first subtitle in integer form (20 four-character strings)
Item 95	The split point of the file
Items 97-98	LONGINT of filesize at write

1.2. Format of the Results File

The next few pages describe the format of the ANSYS results file. (In the following tables, records with a record ID containing an asterisk (*) are those you can read and store into the ANSYS database via the **LDREAD** command.)

Note: The pointers in the solution data headers are relative , not absolute pointers. For example, the 12th item in the solution data header will be relative to a position in the Data Set Index (ptrESL = DSI (*i*) + ptrESL).

This section explains the contents of the results file; that is, those files with the following extensions:

.rfl **.brfl**
.rmg **.brmg**
.rst **.brst**
.rth **.brth**
.lnn

1.2.1. Nomenclature

A load case contains the results for an instance in an analysis. A load case is defined by a load step number and a substep number. A load case is also categorized by a cumulative iteration number and time (or frequency) values. A load case is identified by all three methods in the results file.

The results file does not have to contain all the load cases of an analysis.

A data set is used in this chapter to designate a load case.

For a complex analysis, there will be two data sets for each load case. The first data set contain the real solution and the second contains the imaginary solution.

1.2.2. Standard ANSYS File Header

See Section 1.1.2: The Standard Header for ANSYS Binary Files for a description of this set. File number (Item 1) is 12.

1.2.3. Results File Format

```
*comdeck,fdresu
c *** ansys(r) copyright(c) 2000
c *** ansys, inc

c     **********  description of results file  **********
c     --- used for the following files:
c       .rfl     .brfl
c       .rmg     .brmg
c       .rst     .brst
c       .rth     .brth
c       .lnn(lxx)

      LONGINT         resufpL
      integer         resubk, resuut, resuLong, resuSpare(3)
      common /fdresu/ resufpL, resubk, resuut, resuLong, resuSpare

c   ********** common variable descriptions ***********
co  resufpL      file position on file resu
co  resubk       block number for file resu  (usually 6)
co  resuut       file unit for file resu     (0 if not open)  FUN12
co  resuLong     0, old 32 bit integer form   1, 64 bit form (8.1)

c   See fddesc for documentation of how binary files are stored.

c     **********  file format  **********

c        recid tells the identifier for this record.  Not all records will have
c             identifiers -- they are only indicated for those records whose
c             record pointers are stored in a header.

c        type tells what kind of information is stored in this record:
c             i - integer
c             dp - double precision
c             cmp - complex

c        nrec tells how many records of this description are found here

c        lrec tells how long the records are (how many items are stored)

c recid    type    nrec    lrec     contents

c  ---      i       1      100      standard ANSYS file header (see binhed for
c                                   details of header contents)
```

```
c  ---      i       1       40      .RST FILE HEADER

c                                       12,  maxn,  nnod,resmax,numdof,
c                                     maxe,  nelm,   kan, nsets,ptrend,
c                                    ptrDSI,ptrTIM,ptrLSP,ptrELM,ptrNOD,
c                                    ptrGEO,ptrCYC,CMSflg, csEls, units,
c                                   nSector,csCord,   ptrEnd8   ,fsiflag,
c                                    pmeth,noffst,eoffst,nTrans,ptrTRAN,
c                                    kLong, csNds,cpxrst,     0,     0,
c                                        0,     0,     0,     0,     0

c                                   each item in header is described below:

c                                    fun12  - unit number (resu file is 12)
c                                    maxn   - maximum node number of the model
c                                    nnod   - the actual number of nodes used in
c                                             the solution phase
c                                    resmax - the maximum number of data sets
c                                             allowed on the file (defaults to
c                                             1000; minimum allowed is 10)
c                                    numdof - number of DOFs per node
c                                    maxe   - maximum element number of the
c                                             finite element model
c                                    nelm   - number of finite elements
c                                    kan    - analysis type
c                                    nsets  - number of data sets on the file
c                                    ptrend - pointer to the end of the file
c                                    ptrDSI - pointer to the data steps index
c                                             table
c                                    ptrTIM - pointer to the table of time values
c                                             for a load step
c                                    ptrLSP - pointer to the table of load step,
c                                             substep, and cumulative iteration
c                                             numbers
c                                    ptrELM - pointer to the element equivalence
c                                             table
c                                    ptrNOD - pointer to the nodal equivalence
c                                             table
c                                    ptrGEO - pointer to the beginning of
c                                             geometry information
c                                    ptrCYC - pointer to the table of cyc sym
c                                             nodal-diameters at each load step
c                                    CMSflg - CMS results flag: 0-non cms, >0-cms
c                                    units  - unit system used
c                                             = 0 - user defined units
c                                             = 1 - SI
c                                             = 2 - CSG
c                                             = 3 - British, using feet
c                                             = 4 - British, using inches
c                                             = 6 - MPA
c                                    nSector - number of sectors for cyclic sym
c                                    csCord  - Cyclic symmetry coordinate system
c                                    csEls   - Cyclic sym # eles in master sector
c                                    ptrEnd8  23,24  64 bit file length
c                                    fsiflag - FSI analyis flag
c                                    pmeth   - p-method analyis flag
c                                    noffst  - node offset used in writing file
c                                    eoffst  - elem offset used in writing file
c                                    nTrans  - number of SE transformation vects
c                                    ptrTRAN - pointer to SE transformation vects
c                                    kLong   - 1, 64 bit integer form
c                                    csNds   - Cyclic sym # nds in master sector
c                                   cpxrst   - complex results flag (0-no, 1-yes)

c   ---      i       1     numdof   Degrees of freedom per node
c                                    DOF reference numbers are:
c         UX  = 1, UY  = 2, UZ  = 3, ROTX= 4, ROTY= 5, ROTZ= 6, AX  = 7, AY  = 8
c         AZ  = 9, VX  =10, VY  =11, VZ  =12  ****** 13-18 are spares **********
c         ***************  PRES=19, TEMP=20, VOLT=21, MAG =22, ENKE=23, ENDS=24
c         EMF =25, CURR=26, SP01=27, SP02=28, SP03=29, SP04=30, SP05=31, SP06=32
c                                     (curdof(i),i=1,numdof)
```

```
c   NOD      i       1     nnod     Nodal equivalence table. This table equates
c                                   the number used for storage to the actual
c                                   node number
c                                    (baclst(i),i=1,nnod)

c   ELM      i       1     nelm     Element equivalence table. The ANSYS program
c                                   stores all element data in the numerical
c                                   order that the SOLUTION processor solves the
c                                   elements.  This table equates the order
c                                   number used to the actual element number

c   DSI      i       1    2*resmax  Data sets index table. This record contains
c                                   the record pointers for the beginning of
c                                   each data set. The first resmax records are
c                                   the first 32 bits of the index, the second
c                                   resmax records are the second 32 bits. To
c                                   create the 64 bit pointer, use:
c                                     LONGPTR = largeIntGet (first,second)
c                                   Read the solution data header as follows:
c                                     call bioBasePut (nblk,LONGPTR)
c                                     loc = bioiqr (nblk,12)
c                                     call biord (nblk,loc,...
c                                   The rest of the file reading continues to use
c                                   the ptrXXX's that are in the headers.

c   TIM     dp       1     resmax   Time/freq table. This record contains the
c                                   time (or frequency) values for each data
c                                   set.

c   LSP      i       1    3*resmax  Data set identifiers. This record contains
c                                   the load step, substep, and cumulative
c                                   iteration numbers for each data set.

c   TRAN     dp    nTran     25     Substructure transformation vectors

c   GEO      i       1       40     Geometry data header(was 20 in 32 bit vers)

c                                        0,maxety, maxrl, ndnod,  nelm,
c                                   maxcsy,ptrETY,ptrREL,ptrNOD,ptrCSY,
c                                   ptrEID,     0,     0,     0,     0,
c                                   ptrMAS,csysiz,elmsiz,etysiz, rlsiz,
c                                         ptrETYL,      ptrRELL,
c                                   ptrCSYL,     ptrNODL,      ptrEIDL,
c                                         ptrMASL,     0,     0,     0,
c                                         0,    0,     0,     0,     0

c                                  each item in header is described below:

c                                   0      - position not used
c                                   maxety - the maximum element type reference
c                                            number in the model
c                                   maxrl  - the maximum real constant reference
c                                            number in the model
c                                   ndnod  - the number of defined nodes in the
c                                            model
c                                   nelm   - the number of defined elements in
c                                            the model
c                                   maxcsy - the maximum coordinate system
c                                            reference number in the model
c                                   ptrETY - pointer to the element type index
c                                            table
c                                   ptrREL - pointer to the real constant
c                                            index table
c                                   ptrNOD - pointer to the nodal point
c                                            locations
c                                   ptrCSY - pointer to the local coordinate
c                                            system index table
c                                   ptrEID - pointer to the element index
c                                            table
c                                   0      - position not used
c                                   0      - position not used
c                                   0      - position not used
```

```
c                                   0      - position not used
c                                   ptrMAS - pointer to the diagonal mass matrix
c                                   csysiz - the number of items describing a
c                                            local coordinate system (usually
c                                            24)
c                                   elmsiz - the maximum number of nodes that a
c                                            defined element may have
c                                   etysiz - the number of items describing an
c                                           element type(=IELCSZ from echprm.inc)
c                                   rlsiz  - the maximum number of items
c                                            defining a real constant (0, if no
c                                            real constants are defined)
c                                   ptrETYPL - 64 bit pointer to TYPE
c                                   ptrRELL  - 64 bit pointer to REAL
c                                   ptrCSYL  - 64 bit pointer to CSYS
c                                   ptrNODL  - 64 bit pointer to NODES
c                                   ptrEIDL  - 64 bit pointer to ELEMENTS

c   ETY      i       1     maxety   The element types index table. This record
c                                   contains record pointers for each element
c                                   type description. (Relative to ptrETYPL
c                                   for 64 bit version)

c   ---      i     numety  etysiz   Element type description. Each of these
c                                   records is pointed to by a record pointer
c                                   given in the record labeled ETY. See
c                                   routines echprm and elccmt for a complete
c                                   description of the items stored here.

c                                   These items are typically stored into the
c                                   IELC array, and are used to determine the
c                                   element type characteristics at runtime.
c                                   The following items are typically of
c                                   interest:
c                                   * Item 1     - element type reference number
c                                   * Item 2     - element routine number
c                                   * Items 3-14 - element type option keys
c                                                  (keyopts)
c                                   * Item 34    - DOF/node for this element
c                                                  type.  This is a bit mapping
c                                                  of the DOF/node.
c                                   * Item 61    - number of nodes for this
c                                                  element type (nodelm)
c                                   * Item 63    - number of nodes per element
c                                                  having nodal forces, etc.
c                                                  (nodfor)
c                                   * Item 94    - number of nodes per element
c                                                  having nodal stresses, etc.
c                                                  (nodstr).  This number is the
c                                                  number of corner nodes for
c                                                  higher-ordered elements.

c   REL      i       1     maxrl    Real constants index table.  The record
c                                   contains record pointers for each real
c                                   constant set. (Relative to ptrRELL for
c                                   64 bit version)

c   ---     dp     numrl   varies   Element real constant data.  These records
c                                   contain real constant data used for the
c                                   elements.  (See the ANSYS Elements Reference
c                                   manual for values for a specific element.)
c                                   Each of these records is pointed to by a
c                                   record pointer given in the record labeled
c                                   REL. The length of these records varies for
c                                   each element type (actual length is returned
c                                   from routine BINRD8).

c   CSY      i       1     maxcsy   Coordinate systems index table.  This record
c                                   contains the record pointers for each
c                                   coordinate system set. The ANSYS program
c                                   writes coordinate systems only if local
c                                   coordinate systems were defined.  If a local
```

```
c                                  system was defined, the predefined global
c                                  systems 1 to 2 also will be written.  The
c                                  global Cartesian system 0 will never be
c                                  written. (Relative to ptrCSYSL for 64
c                                  bit version)

c   ---      dp     numcsy  csysiz   Coordinate system description.  These
c                                    records contain coordinate system data for
c                                    each coordinate system defined.  Each of
c                                    these records is pointed to by a record
c                                    pointer given in the record labeled SYS.

c                                    The items stored in each record:

c                                    * Items 1-9  are the transformation matrix.
c                                    * Items 10-12 are the coordinate system
c                                      origin (XC,YC,ZC).
c                                    * Items 13-14 are the coordinate system
c                                      parameters (PAR1, PAR2).
c                                    * Items 16-18 are the angles used to define
c                                      the coordinate system.
c                                    * Items 19-20 are theta and phi singularity
c                                      keys.
c                                    * Item 21 is the coordinate system type
c                                      (0, 1, 2, or 3).
c                                    * Item 22 is the coordinate system reference
c                                      number.

c   NOD      dp       1     7*ndnod  This group contains the node number and
c                                    coordinates (in the order
c                                      Node,X,Y,Z,THXY,THYZ,THZX) for each node.
c                                    (32 bit version)

c   NOD      dp     nnod     7       (64 bit version)
c                                    Node,X,Y,Z,THXY,THYZ,THZX for each node
c                                    Nodes are in the order given by Nodal
c                                    Equivalence Table (baclst(1:nnod))

c   EID      i        1     nelm     Element descriptions index table. This
c                                    record contains the record pointers for each
c                                    element description. (LONGINT (2*nelm) for
c                                    64 bit version, relative to ptrEIDL)

c   ---      i      nelm  10+nodelm  Element descriptions. Each of these records
c                                    is pointed to by a record pointer given in
c                                    the record labeled EID. The length of these
c                                    records varies for each element (actual
c                                    length is returned from routine BINRD8).
c                                    nodelm shown here is the number of nodes for
c                                    this element. Its value is defined in the
c                                    element type description record.

c                                    The items stored in each record:

c                                       mat,  type,  real, secnum, esys,
c                                     death,solidm, shape, elnum,     0,
c                                     NODES

c                                   each item is described below:

c                                     mat    - material reference number
c                                     type   - element type number
c                                     real   - real constant reference number
c                                     secnum - section number
c                                     esys   - element coordinate system
c                                     death  - death flag
c                                              = 0 - alive
c                                              = 1 - dead
c                                     solidm - solid model reference
c                                     shape  - coded shape key
c                                     elnum  - element number
c                                     NODES  - node numbers defining the element.
```

```
c                                          (See the ANSYS Elements Reference
c                                          for nodal order of an element).

c   MAS     dp       1  nnod*numdof  Diagonal mass matrix.



c        The solution information is stored starting at this point in the file.
c        The remaining records on the file are repeated as a group nsets times
c        (once for each data set).  Item nsets is defined in the file header.

c        Each set of data is pointed to by a record pointer given in the record
c        labeled DSI.

c   ---      i       1       200     Solution data header. (was 100 in 32 bit)

c                                    pv3num,  nelm,  nnod,  mask, itime,
c                                      iter,ncumit,   nrf,cs_LSC, nmast,
c                                    ptrNSL,ptrESL, ptrRF,ptrMST, ptrBC,
c                                    rxtrap,  mode,  isym, kcmplx,numdof,
c                                      DOFS,
c                                    positions 51-70 - title,
c                                    positions 71-90 - stitle1,
c                                    dbmtim,dbmdat,dbfncl,soltim,soldat,
c                                    ptrOND,ptrOEL,nfldof,ptrEXA,ptrEXT
c                                    101-102  ptrEXA (was in 99)
c                                    103-104  ptrEXT (was in 100)
c                                    105-106  ptrNSL (was in 11)
c                                    107-108  ptrRF  (was in 13)
c                                    109-110  ptrMST (was in 14)
c                                    111-112  ptrBC  (was in 15)
c                                    113-114  ptrTRF (was in EXT 125)
c                                    115-116  ptrOND (was in 96)
c                                    117-118  ptrOEL (was in 97)
c                                    119-120  ptrESL (was in 12)
c                                    131-132  ptrVSL (was in EXT 196)
c                                    133-134  ptrASL (was in EXT 197)
c                                    135-136         (was in EXT 200)
c                                    137-138         (was in EXT 199)

c                                   each item in header is described below:

c                                    pv3num - current solu set number
c                                    nelm   - number of elements
c                                    nnod   - number of nodes
c                                    mask   - bitmask for the existence of
c                                             several records.  If a bit is set
c                                             here, it indicates that the
c                                             corresponding record exists on the
c                                             file.
c                                             The items in the bitmask that
c                                             correspond to each record are shown
c                                             in the record descriptions below.
c                                    itime  - loadstep
c                                    iter   - iteration number
c                                    ncumit - cumulative iteration number
c                                    nrf    - number of reaction forces
c                                    cs_LSC - cyclic symmetry count of the
c                                             load step for this SOLVE
c                                    nmast  - number of masters
c                                    ptrNSL - pointer to nodal solution
c                                    ptrESL - pointer to element solution
c                                    ptrRF  - pointer to reaction forces
c                                    ptrMST - pointer to the masters
c                                    ptrBC  - pointer to the boundary conditions
c                                    rxtrap - key to extrapolate integration
c                                             point results to nodes
c                                             = 0 - move
c                                             = 1 - extrapolate unless active
c                                                   non-linear
c                                             = 2 - extrapolate always
c                                    mode   - mode number of harmonic loading
```

```
c                                          (for cyclic symmetry: this is cs_LSF
c                                          = first load step for this SOLVE)
c                                    isym   - symmetry for harmonic loading
c                                          (for cyclic symmetry: this is cs_LSL
c                                          = last load step for this SOLVE)
c                                    kcmplx - complex key
c                                          = 0 - real
c                                          = 1 - imaginary
c                                    numdof - number of DOFs/nodes for this data
c                                          set
c                                    DOFS   - DOF/node reference numbers (numdof
c                                          values)
c                                    title  - main title (in integer form)
c                                    stitle1- 1st subtitle (in integer form)
c                                    dbmtim - time (in compact form) when the
c                                          database was last modified
c                                    dbmdat - date (in compact form) when the
c                                          database was last modified
c                                    dbfncl - number of times that the database
c                                          was modified
c                                    soltim - time (in compact form) when the
c                                          solution for this data set was done
c                                    soldat - date (in compact form) when the
c                                          solution for this data set was done
c                                    ptrOND - pointer to the ordered node list
c                                          (load case files only)
c                                    ptrOEL - pointer to the ordered element list
c                                          (load case files only)
c                                    nfldof - number of extra Flotran DOFs/nodes
c                                          for this data set
c                                    ptrEXA - pointer to header extension for
c                                          FLOTRAN DOF/extra DOF list.
c                                    ptrEXT - pointer to header extension
c
c                                    Note: ptrXXX are relative to ptrDSI

c   ---     dp       1        100   Solution header - double precision data

c                                     timfrq,lfacto,lfactn,cptime,  tref,
c                                      tunif, tbulk, volbase, tstep, 0.0,
c                                     velocity-acceleration-center of gravity
c                                      terms (positions 11-28)
c                                     if pmeth=0: load data (positions 51-100)
c                                     if pmeth=1: p convergence values
c                                      (positions 31-100)

c                                    each item is described below:

c                                     timfrq - time value (or frequency value,
c                                              for a modal or harmonic analysis)
c                                     lfacto - the "old" load factor (used in
c                                              ramping a load between old and new
c                                              values)
c                                     lfactn - the "new" load factor
c                                     cptime - elapsed cpu time (in seconds)
c                                     tref   - the reference temperature
c                                     tunif  - the uniform temperature
c                                     tbulk  - Bulk temp for FLOTRAN film coefs.
c                                     VolBase  - Initial total volume for VOF
c                                     tstep  - Time Step size for FLOTRAN analysis
c                                     0.0    - position not used

c                                     positions 11-13 -Linear acceleration terms
c                                     positions 14-16 - Angular velocity
c                                     positions 17-19 - Angular acceleration
c                                     positions 20-22 - Angular velocity about
c                                                       the center of gravity
c                                     positions 23-25 - Angular acceleration
c                                                       about the center of
c                                                       gravity
c                                     positions 26-28 - Center of gravity
c                                                       location
```

```
c                                    if pmeth=1:
c                                       positions 31-100 - P convergence values
c                                    if pmeth=0:
c                                       positions 51-100 - Load data

c                                       position 53 - Convergence key (if 1,
c                                                     substep converged)

c   EXA      i       1         64    Header extension (if ptrEXA=ptrEXT, then
c                                                      ptrEXA is unused.)
c                                    positions   1-32  - current extra Flotran
c                                                      DOFs for this result set
c                                    positions  33-64  - current extra Flotran
c                                                        DOF labels for this set

c                                    Extra Flotran DOF reference numbers are:
c         DENS= 1, VISC= 2, EVIS= 3, COND= 4, ECON= 5, LMD1= 6, LMD2= 7, LMD3= 8
c         LMD4= 9, LMD5=10, LMD6=11, EMD1=12, EMD2=13, EMD3=14, EMD4=15, EMD5=16
c         EMD6=17, PTOT=18, TTOT=19, PCOE=20, MACH=21, STRM=22, HFLU=23, HFLM=24
c         YPLU=25, TAUW=26, SPHT=27, CMUV=28
c         *************************** 29-32 are spares ************************

c   EXT      i       1        200    Header extension
c                                    positions   1-32  - current DOF for this
c                                                        result set
c                                    positions  33-64  - current DOF labels for
c                                                        this result set
c                                    positions  65-84  - The third title, in
c                                                        integer form
c                                    positions  85-104 - The fourth title, in
c                                                        integer form
c                                    positions 105-124 - The fifth title, in
c                                                        integer form
c                                    position  125 - ptrTRF- pointer to FLOTRAN
c                                                    previous time step DOF vals
c                                    position  126 - trnvar- #dof in FLOTRAN
c                                                    prev time st DOF vals.
c                                                    (Note 2 old steps saved,
c                                                    thus #DP is 2*trnvar*nNode)
c                                    position  127 - numvdof, number of velocity
c                                                    items per node (ANSYS
c                                                    transient)
c                                    position  128 - numadof, number of
c                                                    acceleration items per
c                                                    node (ANSYS transient)
c                                    position  131-133 - velocity labels
c                                    position  134-136 - acceleration labels
c                                    position  143 - number of stress items
c                                                    (6 or 11); a -11 indicates
c                                                    to use principles directly
c                                                    and not recompute (for PSD)
c                                    if pmeth=1:
c                                       positions 164-200 - p convergence specs

c * NSL     dp       1  nnod*Sumdof  The DOF solution for each node in the nodal
c                                    coordinate system.  The DOF order is the
c                                    same as shown above in the DOF number
c                                    reference table.  The nodal order is the
c                                    same order given above in the nodal
c                                    equivalence table.  If a DOF for a node
c                                    isn't valid, a value of 2.0**100 is used.
c                                    Note 1: Sumdof = numdof + nfldof.
c                                    Note 2: If, upon reading of this record,
c                                    there is less than nnod*Sumdof items in the
c                                    record, then only a selected set of nodes
c                                    were output. Another record follows
c                                    (integer, less than nnod long) which
c                                    contains the list of nodes for which DOF
c                                    solutions are available.
c                                      (bit 10 (PDBN) in mask)
```

```
c   VSL      dp      1 nnod*numvdof  The velocity solution for each node in the
c                                    nodal coordinate system. The description for
c                                    the DOF solution above also applies here.
c                                    ANSYS transient. (bit 27 (PDVEL) in mask)

c   ASL      dp      1 nnod*numadof  The acceleration solution for each node in
c                                    the nodal coordinate system. The description
c                                    for the DOF solution above also applies here.
c                                    ANSYS transient. (bit 28 (PDACC) in mask)

c   RF       i       1       nrf     Reaction force DOFs.  This index is
c                                    calculated as (N-1)*numdof+DOF, where N is
c                                    the position number of the node in the
c                                    nodal equivalence table, and DOF is the DOF
c                                    reference number.
c                                      (bit 11 (PDBR) in mask)

c * ---     dp       1       nrf     Reaction forces.  The force values are
c                                    ordered according to the DOF order shown
c                                    above in the DOF number reference table.
c                                      (bit 11 (PDBR) in mask)

c   MST      i       1       nmast   Master DOF list.  This index is calculated
c                                    as (N-1)*numdof+DOF, where N is the
c                                    position number of the node in the nodal
c                                    equivalence table, and DOF is the DOF
c                                    reference number.
c                                      (bit 4 in mask)

c   BC       i       1        40     Boundary condition index table.
c                                      (bit 23 (PDBBC) in mask)
c                                    numdis,ptrDIX,ptrDIS,numfor,ptrFIX,
c                                    ptrFOR,     0,     0,     0,     0,
c                                         0,     0,     0,     0,     0,
c                                         0,     0,     0,     0,     0,
c                                         0,     0,     0,     0,     0,
c                                         0,     0,     0,     0,     0,
c                                         0,     0,     0,     0,     0,
c                                         0,     0,     0,     0,     0

c                                  each item is described below:

c                                    numdis - number of nodal constraints
c                                    ptrDIX - pointer to the table of nodes
c                                             having nodal constraints
c                                    ptrDIS - pointer to nodal constraint values
c                                    numfor - number of nodal input force
c                                             loadings
c                                    ptrFIX - pointer to the table of nodes
c                                             having nodal forces
c                                    ptrFOR - pointer to nodal force values

c                                    positions 7-40 are unused.

c   DIX      i       1      numdis   Nodal constraint DOF.  This index is
c                                    calculated as N*32+DOF, where N is the node
c                                    number and DOF is the DOF reference number.
c                                    Values are in the same order as the DOF
c                                    number reference table.

c   DIS     dp       1    4*numdis   Nodal constraints.  This record contains
c                                    present and previous values (real and
c                                    imaginary) of the nodal constraints at each
c                                    DOF.

c   FIX      i       1      numfor   Nodal input force DOFs.  This index is
c                                    calculated as N*32+DOF, where N is the node
c                                    number and DOF is the DOF reference number.
c                                    Values are in the same order as the DOF
c                                    number reference table.

c   FOR     dp       1    4*numfor   Nodal forces.  This record contains present
```

```
c                                    and previous values (real and imaginary) of
c                                    the nodal input force loadings at each DOF.

c   TRF     dp       1    28*nnod    Two displacement result sets for transient
c                                    solution in FLOTRAN
c                                      (bit 24 (PDTRFL) in mask)

c   OND      i       1       nnod    Ordered node list.  This record exists for
c                                    a load case file only.

c   OEL      i       1       nelm    Ordered element list.  This record exists
c                                    for a load case file only.


c   ESL      i       1     2*nelm    Element solutions index table.  This record
c                                    contains pointers to each element solution.
c                                    The order of the elements is the same as
c                                    the order in the element equivalence table.
c                                      (bit 12 (PDBE) in mask)

c        The solution information for each individual element is stored starting
c        at this point in the file.  The next 23 records on the file are
c        repeated as a group nelm times (once for each element).  Item nelm is
c        defined in the file header.

c   ---      i       1       25      Individual element index table.

c                                    ptrEMS,ptrENF,ptrENS,ptrENG,ptrEGR,
c                                    ptrEEL,ptrEPL,ptrECR,ptrETH,ptrEUL,
c                                    ptrEFX,ptrELF,ptrEMN,ptrECD,ptrENL,
c                                    ptrEHC,ptrEPT,ptrESF,     0,ptrETB,
c                                    ptrECT,ptrEXY,ptrEBA,ptrESV,     0
c                                   (Relative to ptrESL for 64 bit version)

c                                  each item is described below:

c                                    ptrEMS - pointer to misc. data
c                                    ptrENF - pointer to nodal forces
c                                    ptrENS - pointer to nodal stresses
c                                    ptrENG - pointer to volume and energies
c                                    ptrEGR - pointer to nodal gradients
c                                    ptrEEL - pointer to elastic strains
c                                    ptrEPL - pointer to plastic strains
c                                    ptrECR - pointer to creep strains
c                                    ptrETH - pointer to thermal strains
c                                    ptrEUL - pointer to euler angles
c                                    ptrEFX - pointer to nodal fluxes
c                                    ptrELF - pointer to local forces
c                                    ptrEMN - pointer to misc. non-sum values
c                                    ptrECD - pointer to element current
c                                             densities
c                                    ptrENL - pointer to nodal nonlinear data
c                                    ptrEHC - pointer to calculated heat
c                                             generations
c                                    ptrEPT - pointer to element temperatures
c                                    ptrESF - pointer to element surface
c                                             stresses
c                                    ptrETB - pointer to ETABLE items(post1 only
c                                    ptrECT - pointer to contact data
c                                    ptrEXY - pointer to integration point locations
c                                    ptrEBA - pointer to back stresses
c                                    ptrESV - pointer to state variables
c                                    0      - position not used

c                                    Note! If ptrXXX is negative, then all |ptrXXX|
c                                    items are zero and are not on the file.

c   EMS     dp       1    varies     Element summable miscellaneous data.  The
c                                    contents and number of data items is
c                                    element-dependent.  For a list of what's
c                                    available, see the SMISC item in the
c                                    description of the ETABLE command in the
```

```
c                                       ANSYS Commands Reference.

c   ENF     dp       1    varies      Element nodal forces.  This record contains
c                                       the forces at each node, in the same DOF
c                                       order as the DOF number reference table.
c                                       For static, damping, and inertia forces, a
c                                       set of forces will be repeated (as
c                                       appropriate).  Number of data items stored
c                                       in this record can be calculated as
c                                       follows: nodfor*NDOF*M, where NDOF is the
c                                       number of DOFs/node for this element,
c                                       nodfor is the number of nodes per element
c                                       having nodal forces (defined in element
c                                       type description record), and M may be 1,
c                                       2, or 3.  For a static analysis, M=1 only.
c                                       For a transient analysis, M can be 1, 2,
c                                       or 3.

c   ENS     dp       1    varies      Element nodal component stresses.  This
c                                       record contains the stresses at each corner
c                                       node, in the order SX,SY,SZ,SXY,SYZ,SXZ,S1,
c                                       S2,S3,SI,SIGE.  Nodal order corresponds to
c                                       the connectivity defined in the element
c                                       description.  Stresses can be nodal values
c                                       extrapolated from the integration points or
c                                       values at the integration points moved to
c                                       the nodes.  If an element is nonlinear,
c                                       integration point values always will be
c                                       written.  (See item rxtrap in the solution
c                                       header for the setting.)  An element is
c                                       considered nonlinear when either plastic,
c                                       creep, or swelling strains are present.

c                                       * For solid elements, stresses are at each
c                                         corner node and the number of items in
c                                         this record is nodstr*11.
c                                       * For shell elements, stresses are at each
c                                         corner node (first at the top surface,
c                                         then the bottom) and the number of items
c                                         in this record is 2*nodstr*11.
c                                       * For layered elements (with KEYOPT(8)=0),
c                                         stresses for the "first" layer are at
c                                         each corner node (first at the bottom
c                                         surface of the bottom layer, then at the
c                                         top surface of the top layer).  Stresses
c                                         for the "second" layer are at each corner
c                                         node (first at the bottom surface, then
c                                         at the top surface for the layer with the
c                                         largest failure criteria).  The number of
c                                         items in this record is NL*2*nodstr*11,
c                                         where NL is 2 (for two layers).  The
c                                         second layer isn't present if failure
c                                         criteria weren't used or aren't
c                                         appropriate.
c                                       * For layered elements (with KEYOPT(8)=1),
c                                         stresses for each layer are at each
c                                         corner
c                                         node (first at the bottom surface, then
c                                         at the top surface) and the number of
c                                         items in this record is NL*2*nodstr*11,
c                                         where NL is the number of layers.
c                                       * For beam elements, the contents and number
c                                         of data items is element-dependent.  See
c                                         the Output Data section for the particular
c                                         element in the ANSYS Elements Reference.

c                                       NOTE: nodstr is defined in the element type
c                                       description record.

c   ENG     dp       1      11        Element volume and energies.

c                                       volume,senergy,aenergy,kenergy,coenergy,
```

```
c                                   incenergy,0.0,0.0,thenergy,0.0,0.0

c                                  each item is described below:

c                                   volume   - element volume
c                                   senergy  - element energy associated with
c                                              the stiffness matrix
c                                   aenergy  - artificial hourglass energy
c                                   kenergy  - kinetic energy
c                                   coenergy - co-energy (magnetics)
c                                   incenergy- incremental energy (magnetics)
c                                   0.0      - position not used
c                                   0.0      - position not used
c                                   thenergy - thermal dissipation energy
c                                              (see ThermMat, shell131/132 only)
c                                   0.0      - position not used
c                                   0.0      - position not used

c   EGR     dp       1    varies     Element nodal field gradients.  This record
c                                    contains the gradients at each corner node
c                                    in the order X,Y,Z.  Nodal order
c                                    corresponds to the connectivity defined in
c                                    the element description.  If this is a
c                                    coupled-field analysis, the data is stored
c                                    in the following order (as available):
c                                    fluid, thermal (TEMP), electric (VOLT), and
c                                    magnetic (AZ).  Gradients can be nodal
c                                    values extrapolated from the integration
c                                    points or values at the integration points
c                                    moved to the nodes.  See item rxtrap in the
c                                    solution header for the setting.  The
c                                    number of items in this record is
c                                    nodstr*3*N, where N can be 1, 2, 3, or 4
c                                    (depending on the coupled-field
c                                    conditions).

c                                    NOTE: nodstr is defined in the element type
c                                    description record.

c   EEL     dp       1    varies     Element nodal component elastic strains.
c                                    This record contains strains in the order
c                                    X,Y,Z,XY,YZ,XZ,EQV. Elastic strains can be
c                                    can be nodal values extrapolated from the
c                                    integration points or values at the
c                                    integration points moved to the nodes.  If
c                                    an element is nonlinear, integration point
c                                    values always will be written.  See item
c                                    rxtrap in the solution header for the
c                                    setting.  An element is considered
c                                    nonlinear when either plastic, creep, or
c                                    swelling strains are present.  For beam
c                                    elements, see item LEPEL in the description
c                                    in the Output Data section for the particular
c                                    element in the ANSYS Elements Reference.

c   EPL     dp       1    varies     Element nodal component plastic strains.
c                                    This record contains strains in the order
c                                    X,Y,Z,XY,YZ,XZ,EQV.
c                                    Plastic strains are always values at the
c                                    integration points moved to the nodes.  For
c                                    beam elements, see item LEPPL in the
c                                    Output Data section for the particular
c                                    element in the ANSYS Elements Reference.

c   ECR     dp       1    varies     Element nodal component creep strains.
c                                    This record contains strains in the order
c                                    X,Y,Z,XY,YZ,XZ,EQV.
c                                    Creep strains are always values at the
c                                    integration points moved to the nodes.  For
c                                    beam elements, see item LEPCR in the
c                                    Output Data section for the particular
c                                    element in the ANSYS Elements Reference.
```

```
c   ETH     dp       1    varies     Element nodal component thermal strains.
c                                    This record contains strains in the order
c                                    X,Y,Z,XY,YZ,XZ,EQV plus the element
c                                    swelling strain. Thermal
c                                    strains can be nodal values extrapolated
c                                    from the integration points or values at
c                                    the integration points moved to the nodes.
c                                    If the element in nonlinear, integration
c                                    point data always will be written. (An
c                                    element is considered nonlinear when either
c                                    plastic, creep, or swelling strains are
c                                    present.) See item rxtrap in the solution
c                                    header for the setting.  For beam elements,
c                                    see item LEPTH in the description of the
c                                    Output Data section for the particular
c                                    element in the ANSYS Elements Reference.

c   EUL     dp       1    varies     Element Euler angles.  This record contains
c                                    the Euler rotations (THXY,THYZ,THZX).

c                                    * For lower-ordered elements, rotations are
c                                      at the centroid and the number of items
c                                      in this record is 3.
c                                    * For higher-ordered elements, rotations
c                                      are at each corner node and the number of
c                                      items in this record is nodstr*3.
c                                    * For layered shells, rotations are at each
c                                      corner node, plus the layer rotation
c                                      angle for each layer (real constant
c                                      THETA).  The number of items in this
c                                      record is nodstr*3+NL, where NL is the
c                                      number of layers.
c                                    * For the layered solid, rotation angles
c                                      are at the centroid, plus the layer
c                                      rotation angle for each layer (real
c                                      constant THETA).  The number of items in
c                                      this record is 3 + NL.

c                                    NOTE: nodstr is defined in the element type
c                                    description record.

c   EFX     dp       1    varies     Element nodal field fluxes.  This record
c                                    contains the fluxes at each corner node in
c                                    the order X,Y,Z. If this is a
c                                    coupled-field analysis, the flux data is
c                                    stored in the following order:  thermal,
c                                    electric, magnetic.  Nodal order
c                                    corresponds to the connectivity defined in
c                                    the element description.  Fluxes can be
c                                    nodal values extrapolated from the
c                                    integration points or values at the
c                                    integration points moved to the nodes.
c                                    See item rxtrap in the solution header for
c                                    the setting.  The number of items in this
c                                    record is nodstr*3*N, where N can be 1, 2,
c                                    or 3 depending on the coupled-field
c                                    conditions.

c                                    NOTE: nodstr is defined in the element type
c                                    description record.

c * ELF     dp       1    varies     Element nodal coupled-field forces.  This
c                                    record lists the forces at each node in the
c                                    order X,Y,Z.  For most elements, the number
c                                    of items in this record is nodfor*3.
c                                    However, for the PLANE53 element, the
c                                    number of items in this record is either
c                                    nodfor*3 or nodstr*3.  (See the description
c                                    of KEYOPT(7) for PLANE53 in the ANSYS
c                                    Elements Reference.)  NOTE: nodfor and
c                                    nodstr are defined in the element type
```

```
c                                   description record.

c                                   NOTE: nodstr is defined in the element type
c                                   description record.

c   EMN     dp       1    varies     Element nonsummable miscellaneous data.
c                                   The contents and number data items for this
c                                   record is element-dependent.   See the
c                                   description for item NMISC of the ETABLE
c                                   command in the ANSYS Commands Reference.

c * ECD     dp       1       3       Element current densities.  This record
c                                   contains the calculated current densities
c                                   in the order X,Y,Z.

c   ENL     dp       1    varies     Element nodal nonlinear data.  This record
c                                   stores nonlinear data at each corner node
c                                   in the order SEPL, SRAT, HPRES, EPEQ, PSV,
c                                   PLWK, CRWK, and ELENG followed by 2 spares.

c                                  each item is described below:
c                                     SEPL  - equivalent stress parameter
c                                     SRAT  - stress ratio
c                                     HPRES - hydrostatic pressure
c                                     EPEQ  - accumulated equivalent plastic strain
c                                     PSV   - plastic state variable
c                                     PLWK  - plastic strain energy density (work)
c                                     CRWK  - creep strain energy density (work)
c                                     ELENG - elestic strain energy density

c                                     * For solid elements, the record contains
c                                       nonlinear data at each corner node and
c                                       the number of items in this record is
c                                       nodstr*10.
c                                     * For shell elements, the record contains
c                                       nonlinear data at each corner node (first
c                                       at the top surface, them the bottom
c                                       surface), and the number of items in this
c                                       record is 2*nodstr*10.
c                                     * For layered elements, the record contains
c                                       nonlinear data at each layer at each
c                                       corner node (first at the top surface,
c                                       then the bottom) and the number of items
c                                       in this record is NL*2*nodstr*10, where
c                                       NL = number of layers.
c                                     * For beam elements, the contents and
c                                       number of data items in this record is
c                                       element-dependent.  See the description
c                                       of item NLIN in the Output Data section
c                                       for the particular element in the ANSYS
c                                       Elements Reference.

c                                     NOTE: nodstr is defined in the element type
c                                     description record.

c * EHC     dp       1       1       Element heat generation.  This record
c                                   stores the calculated heat generation.

c   EPT     dp       1    varies     Element structural nodal temperatures.

c                                     * For solid elements and SHELL41, the
c                                       record contains nodal temperatures at
c                                       each node and the number of items in this
c                                       record is nodfor.
c                                     * For shell elements, except SHELL41 and
c                                       SHELL91, the record contains nodal
c                                       temperatures at each corner node for the
c                                       top surface and the bottom surface.  The
c                                       number of items in this record is
c                                       nodstr*2.
c                                     * For SHELL91 and SOLID191, the record
c                                       contains nodal temperatures at each
```

```
c                                     corner node for the bottom of the bottom
c                                     layer, and each succeeding interlayer
c                                     surface up to the top of the top layer.
c                                     The number of items in this record is
c                                     (NL+1)*nodstr, where NL = number of
c                                     layers.
c                                   * For beam elements, the contents and
c                                     number of data items in this record is
c                                     element-dependent.  See the description
c                                     of item LBFE in the Output Data section
c                                     for the particular element in the ANSYS
c                                     Elements Reference.

c                                   NOTE: nodfor and nodstr are defined in the
c                                   element type description record.

c   ESF     dp       1    nsurf*19   Element surface stresses.  The
c                                     length of this record is nsurf*19 where
c                                     nsurf is the number of surfaces that have
c                                     surface stress information.  The stress
c                                     information is simply repeated in the
c                                     format shown below for each surface.

c                                     * For 2d elements:

c                                     facenm,  area,  temp, press, eppar,
c                                      epper,   epz, 0.0d0,  spar,  sper,
c                                         sz, 0.0d0, 0.0d0, 0.0d0,    s1,
c                                         s2,    s3,  sint,  seqv

c                                     * For 3d elements:

c                                     facenm,  area,  temp, press,   epx,
c                                        epy,   epz,  epxy,    sx,    sy,
c                                         sz,   sxy, 0.0d0, 0.0d0,    s1,
c                                         s2,    s3,  sint,  seqv

c                                     * For axisymmetric elements:

c                                     facenm,  area,  temp, press, eppar,
c                                      epper,   epz,  epsh,  spar,  sper,
c                                         sz, 0.0d0, 0.0d0,   ssh,    s1,
c                                         s2,    s3,  sint,  seqv

c                                   each item is described below:

c                                     facenm - face number
c                                     area   - face area
c                                     temp   - face temperature
c                                     press  - face pressure
c                                     epx    - strain parallel to face
c                                     epy    - strain parallel to face
c                                     epz    - strain perpendicular to face
c                                     epxy   - shear strain
c                                     eppar  - strain parallel to face
c                                     epper  - strain perpendicular to face
c                                     epsh   - torsion shear strain
c                                     sx     - stress parallel to face
c                                     sy     - stress parallel to face
c                                     sz     - stress perpendicular to face
c                                     sxy    - shear stress
c                                     spar   - stress parallel to face
c                                     sper   - stress perpendicular to face
c                                     ssh    - torsion shear stress
c                                     s1     - S(1)
c                                     s2     - S(2)
c                                     s3     - S(3)
c                                     sint   - S(INT)
c                                     seqv   - S(EQV)
c                                     0.0d0  - position not used
```

```
c   EXY     dp       1    varies     Element integration point coordinates
c                                    The length of the record is numint*3, where
c                                    numint is the number of integration points.
c                                    Even two-dimensional elements use the 3.
c                                    They are output only if requested with the
c                                    OUTRES,loci command.
c                                    Applicable only to element types 2,42,45,
c                                    82,92,95, and 180 to 187.

c   EBA     dp       1    varies     Element structural nodal back stresses
c                                    Record has the same form as the plastic
c                                    strains.  They are output if the form of
c                                    plasticity is kinematic hardening and the
c                                    plastic strains are requested.
c                                    Applicable only to element types 2,42,45,
c                                    82,92,95, and 180 to 187.

c   ESV     dp       1    varies     Element state variable record.  Exists only
c                                    if written by user in usermat or usercreep.

c   records marked with * to the left of the record id can be read and stored
c   into database with "ldread" command.
```

1.3. Description of the Reduced Displacement File

This section explains the content of the reduced displacement file (**jobname.rdsp**).

1.3.1. Standard ANSYS File Header

See Section 1.1.2: The Standard Header for ANSYS Binary Files for a description of this set. File number (Item 1) is 10.

1.3.2. RDSP File Format

```
*comdeck,fdrdsp
c *** ansys(r) copyright(c) 2000
c *** ansys, inc


c     **********  description of reduced displacement file  **********
      character*8  RDSPNM
      parameter  (RDSPNM='rdsp    ')

      LONGINT         rdspfpL, rdspfp
      integer         rdspbk,  rdsput
      common /fdrdsp/ rdspfpL, rdspbk, rdsput
      equivalence  (rdspfp,rdspfpL)

c  write:  lnfrcl,lnfrin,lnfrwr
c  write:  rdtrcl,rdtrin,rdtrwr
c  read:   rdtrrs,rdtrs

c   ********** common variable descriptions ***********
co  rdspfpL      file position on file rdsp
co  rdspbk       block number for file rdsp
co  rdsput       file unit for file rdsp

c   See fddesc for documentation of how binary files are stored.
c
c     **********  file format  **********

c        recid tells the identifier for this record.  Not all records will have
c             identifiers -- they are only indicated for those records whose
c             record pointers are stored in the second file header.

c        type tells what kind of information is stored in this record:
c             i - integer
```

```
c             dp - double precision
c             cmp - complex

c        nrec tells how many records of this description are found here

c        lrec tells how long the records are (how many items are stored)

c recid    type    nrec    lrec     contents

c  ---      i       1      100      standard ANSYS file header (see binhed for
c                                    details of header contents)

c  ---      i       1       40    .RDSP FILE HEADER
c
c                                    fun10,  nmrow, nmatrx,  nmode, numdof,
c                                     maxn,  wfmax, lenbac,  ngaps, ncumit,
c                                       kan,      0,      0,      0,      0,
c                                         0,      0,      0,      0,      0,
c                                   ptrDOF, ptrDNC, ptrSTF, ptrMAS, ptrDMP,
c                                   ptrFRQ, ptrDSP,ptrSTFh,ptrMASh,ptrDMPh,
c                                  ptrFRQh,ptrDSPh,      0,      0,      0,
c                                        0,      0,      0,      0,      0

c                                  each item in header is described below:

c                                   fun10  - unit number (rdsp file is 10)
c                                   nmrow  - number of rows/columns in matrices
c                                   nmatrx - number of reduced matrices on the
c                                            file
c                                   nmode  - number of modes extracted during
c                                            modal analysis (or nmrow if reduced
c                                            method)
c                                   numdof - number of dofs per node
c                                   maxn   - maximum node number
c                                   wfmax  - maximum wavefront
c                                   lenbac - number of nodes
c                                   ngaps  - number of gaps
c                                   ncumit - total number of iterations done
c                                            during analysis
c                                   kan    - analysis type
c                                            = 5 for reduced transient analysis
c                                   0      - position not used
c                                   0      - position not used
c                                   0      - position not used
c                                   0      - position not used
c                                   0      - position not used
c                                   0      - position not used
c                                   0      - position not used
c                                   0      - position not used
c                                   0      - position not used
c                                   ptrDOF - pointer to degree of freedom set
c                                   ptrDNC - pointer to nodal constraints
c                                   ptrSTF - pointer to the reduced stiffness
c                                   ptrMAS - pointer to the reduced mass matrix
c                                   ptrDMP - pointer to the reduced damping
c                                            matrix or mode shapes
c                                   ptrFRQ - pointer to the frequencies
c                                   ptrDSP - pointer to the calculated
c                                            displacements
c                                   ptrSTFh- High part of reduced stiffness ptr
c                                   ptrMASh- High part of reduced mass ptr
c                                   ptrDMPh- High part of reduced damping ptr
c                                   ptrFRQh- High part of frequency ptr
c                                   ptrDSPh- High part of displacement ptr
c                                   0      - position not used
c                                   0      - position not used
c                                   0      - position not used
c                                   0      - position not used
c                                   0      - position not used
c                                   0      - position not used
c                                   0      - position not used
c                                   0      - position not used
```

```
c  ---      i       1     numdof  Degrees of freedom per node
c                                 (curdof(i),i=1,numdof)
c                                dof reference numbers are:
c         UX  = 1, UY  = 2, UZ  = 3, ROTX= 4, ROTY= 5, ROTZ= 6, AX  = 7, AY  = 8
c         AZ  = 9, VX  =10, VY  =11, VZ  =12  ****** 13-18 are spares **********
c         ***************  PRES=19, TEMP=20, VOLT=21, MAG =22, ENKE=23, ENDS=24
c         EMF =25, CURR=26  ********* 27-32 are spares ************************

c  ---      i       1     lenbac  This table equates the actual node number to
c                                 the number used for storage.
c                                  (baclst(i),i=1,lenbac)

c  ---      dp      1       10    Time information:
c                                  dtime,    0.0,    0.0,    0.0,    0.0,
c                                    0.0,    0.0,    0.0,    0.0, timend

c                                each item is described below:
c
c                                  dtime  - the time increment
c                                  0.0    - position not used
c                                  0.0    - position not used
c                                  0.0    - position not used
c                                  0.0    - position not used
c                                  0.0    - position not used
c                                  0.0    - position not used
c                                  0.0    - position not used
c                                  timend - the final time of the analysis

c  DOF      i       1     nmrow    Degree of freedom set used
c                                  The DOFs are calculated as (N-1)*numdof+DOF,
c                                  where N is the position number of the node in
c                                  the nodal equivalence table and DOF is the
c                                  DOF reference number given above.

c                                  If the analysis uses the reduced method, the
c                                  original DOF order (see next record) is
c                                  rearranged so that DOFs having nodal
c                                  constraints are listed first.
c
c                                  If the analysis uses the mode superposition
c                                  method (using the reduced mode extraction
c                                  technique), the DOF order is the same as the
c                                  original order (see next record).
c                                   (l(i),i=1,nmrow)

c  ---      i       1     nmrow+1  Original reduced set of DOFs used.
c                                  The DOFs are calculated as (N-1)*numdof+DOF,
c                                  where N is the position number of the node in
c                                  the nodal equivalence table and DOF is the
c                                  DOF reference number given above.

c                                  If the analysis uses the reduced method, the
c                                  original DOF order, plus the number of nodal
c                                  constraints (nbcdsp), is stored.

c                                  If the analysis uses the mode superposition
c                                  method (using the reduced mode extraction
c                                  technique), this record matches the previous
c                                  record.  The nmrow+1 entry will be zero.
c                                   (lorig(i),i=1,nmrow),nbcdsp

c  DNC      i       1     nbcdsp   This record is present only if the analysis
c                                  uses the reduced method and nbcdsp > 0 (see
c                                  record at ptrDOF). These numbers are the
c                                  positions in the previous record of dofs with
c                                  a nodal constraint. These are nodal
c                                  constraints only on nodes that also are
c                                  masters.
c                                   (na(i),i=1,nbcdsp)
```

```
c  STF      dp    nmrow   nmrow    Reduced stiffness matrix. Each row of the
c                                   matrix is stored as a record. The matrix is
c                                   present only if nmatrx > 0 and analysis is
c                                   not using mode superposition method (using
c                                   the subspace mode extraction method). Row
c                                   order is the same as the DOF order in record
c                                   at ptrDOF.
c                                    (ak(i,j),i=1,nmrow)

c  MAS      dp    nmrow   nmrow    Reduced mass matrix. Each row of the matrix
c                                   is stored as a record. The matrix is present
c                                   only if nmatrx > 1 and analysis is not using
c                                   mode superposition method (using the subspace
c                                   extraction technique). Row order is the same
c                                   as the DOF order in record at ptrDOF.
c                                    (am(i,j),i=1,nmrow)

c  DMP      dp     varies  varies  Reduced damping matrix or mode shapes.

c                                   If the analysis uses the reduced method,
c                                   each record will be nmrow items in length.
c                                   The reduced damping matrix is present only
c                                   if nmatrx > 2. There will be nmrow records of
c                                   this type stored here. Row order is the same
c                                   as the DOF order in record at ptrDOF.

c                                   If the analysis uses the mode superposition
c                                   method (using the reduced mode extraction
c                                   technique), each record will be nmode items
c                                   in length. These records contain mode shapes
c                                   (eigenvectors) of the frequencies
c                                   (eigenvalues) actually used in the harmonic
c                                   analysis. There will be nmode records of this
c                                   type stored here, with the first N records
c                                   containing the mode shapes and the other
c                                   records containing zeros, where N is the
c                                   number of modes actually used in the harmonic
c                                   analysis. Order corresponds to the DOF order
c                                   given in record at ptrDOF.

c                                   If the analysis uses the mode superposition
c                                   method (using the subspace mode extraction
c                                   technique), this record will not be present.
c                                     (psi(i,j),i=1,nmrow) (or ac)

c  FRQ      dp      1     nmrow    Frequencies extracted from the modal
c                                   analysis. This record is present only if the
c                                   analysis uses the mode superposition method.
c                                   The first nmode values are the frequencies
c                                   extracted from the modal analysis.  The
c                                   remaining values have no meaning.
c                                    (freq(i),i=1,nmrow)

c  *** The next 2 records are repeated (as a pair) until the time value
c  *** equals the value of timend.  The number of iterations is stored as
c  *** ncumit. (see above records that deal with time)

c  DSP      dp      1     nmrow+5  Calculated displacements
c                                   The first nmrow entries are the displacements
c                                   in the same order as the original set of DOFs
c                                   (see record AFTER ptrDOF). For the last five
c                                   entries:
c                                   1. Time for these displacements
c                                   2. Load step number
c                                   3. Substep number
c                                   4. Cumulative iteration number
c                                   5. Scale factor (zero if the analysis uses
c                                      the reduced method).
c                                       (u(i),i=1,nmrow),time,itime,iter,ncumit,
c                                       scale
c                                   Note: If, upon reading of this record, there
c                                   is less than nmrow+5 items in the record,
```

```
c                                then only a selected set of nodes were
c                                output. Another record follows (integer, less
c                                than lenbac long) which contains the list of
c                                nodes for which DOF solutions are available.

c  ---      dp      1     ngaps    Gap restoring forces. The order of these
c                                  forces corresponds to the node position order
c                                  given in record at ptrDNC. This record is
c                                  present only if the analysis uses the reduced
c                                  method and ngaps > 0.
c                                   (fgaps(i),i=1,ngaps)
```

1.4. Description of the Reduced Complex Displacement File

This section explains the content of the reduced complex displacement file (**jobname.rfrq**).

1.4.1. Standard ANSYS File Header

See Section 1.1.2: The Standard Header for ANSYS Binary Files for a description of this set. File number (Item 1) is 10.

1.4.2. RFRQ File Format

```
*comdeck,fdrfrq
c *** ansys(r) copyright(c) 2000
c *** ansys, inc


c     **********  description of reduced complex displacement file  **********
      character*8  RFRQNM
      parameter  (RFRQNM='rfrq    ')

      LONGINT         rfrqfpL, rfrqfp
      integer         rfrqbk,  rfrqut
      common /fdrfrq/ rfrqfpL, rfrqbk, rfrqut
      equivalence  (rfrqfp,rfrqfpL)

c  write:  harmcl,harmin,harmwr
c  write:  hrfrcl,hrfreq
c  read:   harstr

c   ********** common variable descriptions ***********
co  rfrqfpL      file position on file rfrq
co  rfrqbk       block number for file rfrq
co  rfrqut       file unit for file rfrq

c   See fddesc for documentation of how binary files are stored.
c
c     **********  file format  **********

c        recid tells the identifier for this record.  Not all records will have
c             identifiers -- they are only indicated for those records whose
c             record pointers are stored in the second file header.

c        type tells what kind of information is stored in this record:
c             i - integer
c             dp - double precision
c             cmp - complex

c        nrec tells how many records of this description are found here

c        lrec tells how long the records are (how many items are stored)

c recid    type    nrec    lrec     contents

c  ---      i       1      100      standard ANSYS file header (see binhed for
c                                   details of header contents)
```

```
c  ---      i       1       40    .RFRQ FILE HEADER
c
c                                    fun10,  nmrow, nmatrx,  nmode, numdof,
c                                     maxn,  wfmax, lenbac,      0, ncumit,
c                                       kan,      0,      0,      0,      0,
c                                         0,      0,      0,      0,      0,
c                                   ptrDOF, ptrDNC, ptrSTF, ptrMAS, ptrDMP,
c                                   ptrFRQ, ptrDSP,ptrSTFh,ptrMASh,ptrDMPh,
c                                  ptrFRQh,ptrDSPh,      0,      0,      0,
c                                        0,      0,      0,      0,      0

c                                  each item in header is described below:

c                                   fun10  - unit number (rfrq file is 10)
c                                   nmrow  - number of rows/columns in matrices
c                                   nmatrx - number of reduced matrices on file
c                                   nmode  - number of modes extracted during
c                                            modal analysis (or nmrow if reduced
c                                            method)
c                                   numdof - number of dofs per node
c                                   maxn   - maximum node number
c                                   wfmax  - maximum wavefront
c                                   lenbac - number of nodes
c                                   0      - position not used
c                                   ncumit - total number of iterations done
c                                            during analysis
c                                   kan    - analysis type
c                                            = 6  - reduced harmonic
c                                   0      - position not used
c                                   0      - position not used
c                                   0      - position not used
c                                   0      - position not used
c                                   0      - position not used
c                                   0      - position not used
c                                   0      - position not used
c                                   0      - position not used
c                                   0      - position not used
c                                   ptrDOF - pointer to degree of freedom set
c                                            used in model
c                                   ptrDNC - pointer to nodal constraints
c                                   ptrSTF - pointer to the reduced stiffness
c                                            matrix
c                                   ptrMAS - pointer to the reduced mass matrix
c                                   ptrDMP - pointer to the reduced damping
c                                            matrix or mode shapes
c                                   ptrFRQ - pointer to the frequencies
c                                   ptrDSP - pointer to the calculated
c                                            displacements
c                                   ptrSTFh- High part of STF pointer
c                                   ptrMASh- High part of MAS pointer
c                                   ptrDMPh- High part of DMP pointer
c                                   ptrFRQh- High part of FRQ pointer
c                                   ptrDSPh- High part of DSP pointer
c                                   0      - position not used
c                                   0      - position not used
c                                   0      - position not used
c                                   0      - position not used
c                                   0      - position not used
c                                   0      - position not used
c                                   0      - position not used
c                                   0      - position not used

c  ---      i       1     numdof   Degrees of freedom per node
c                                    (curdof(i),i=1,numdof)
c                                   dof reference numbers are:
c        UX  = 1, UY  = 2, UZ  = 3, ROTX= 4, ROTY= 5, ROTZ= 6, AX  = 7, AY  = 8
c        AZ  = 9, VX  =10, VY  =11, VZ  =12  ****** 13-18 are spares **********
c        ***************  PRES=19, TEMP=20, VOLT=21, MAG =22, ENKE=23, ENDS=24
c        EMF =25, CURR=26  ********* 27-32 are spares ************************

c  ---      i       1     lenbac  This table equates the actual node number to
```

```
c                                   the number used for storage.
c                                    (baclst(i),i=1,lenbac)

c  ---      dp      1       10     Unused record. contents:
c                                   1.0, 0.0, 0.0, 0.0, 0.0,
c                                   0.0, 0.0, 0.0, 0.0, 0.0

c  DOF      i       1     nmrow     Degree of freedom set used
c                                   The DOFs are calculated as (N-1)*numdof+DOF,
c                                   where N is the position number of the node in
c                                   the nodal equivalence table and DOF is the
c                                   DOF reference number given above.

c                                   If the analysis uses the reduced method, the
c                                   original DOF order (see next record) is
c                                   rearranged so that DOFs having nodal
c                                   constraints are listed first.
c
c                                   If the analysis uses the mode superposition
c                                   method (using the reduced mode extraction
c                                   technique), the DOF order is the same as the
c                                   original order (see next record).
c                                    (l(i),i=1,nmrow)

c  ---      i       1     nmrow+1   Original reduced set of DOFs used.
c                                   The DOFs are calculated as (N-1)*numdof+DOF,
c                                   where N is the position number of the node in
c                                   the nodal equivalence table and DOF is the
c                                   DOF reference number given above.

c                                   If the analysis uses the reduced method, the
c                                   original DOF order, plus the number of nodal
c                                   constraints (nbcdsp), is stored.

c                                   If the analysis uses the mode superposition
c                                   method (using the reduced mode extraction
c                                   technique), this record matches the previous
c                                   record.  The nmrow+1 entry will be zero.
c                                    (lorig(i),i=1,nmrow),nbcdsp

c  DNC      i       1     nbcdsp    This record is present only if the analysis
c                                   uses the reduced method and nbcdsp > 0 (see
c                                   record at ptrDOF). These numbers are the
c                                   positions in the previous record of dofs with
c                                   a nodal constraint. These are nodal
c                                   constraints only on nodes that also are
c                                   masters.
c                                    (na(i),i=1,nbcdsp)

c  STF      dp    nmrow   nmrow     Reduced stiffness matrix. Each row of the
c                                   matrix is stored as a record. The matrix is
c                                   present only if nmatrx > 0 and analysis is
c                                   not using mode superposition method (using
c                                   the subspace mode extraction method). Row
c                                   order is the same as the DOF order in record
c                                   at ptrDOF.
c                                    (ak(i,j),i=1,nmrow)

c  MAS      dp    nmrow   nmrow     Reduced mass matrix. Each row of the matrix
c                                   is stored as a record. The matrix is present
c                                   only if nmatrx > 1 and analysis is not using
c                                   mode superposition method (using the subspace
c                                   extraction technique). Row order is the same
c                                   as the DOF order in record at ptrDOF.
c                                    (am(i,j),i=1,nmrow)

c  DMP      dp     varies  varies   Reduced damping matrix or mode shapes.

c                                   If the analysis uses the reduced method,
c                                   each record will be nmrow items in length.
c                                   The reduced damping matrix is present only
c                                   if nmatrx > 2. There will be nmrow records of
```

```
c                                this type stored here. Row order is the same
c                                as the DOF order in record at ptrDOF.

c                                If the analysis uses the mode superposition
c                                method (using the reduced mode extraction
c                                technique), each record will be nmode items
c                                in length. These records contain mode shapes
c                                (eigenvectors) of the frequencies
c                                (eigenvalues) actually used in the harmonic
c                                analysis. There will be nmode records of this
c                                type stored here, with the first N records
c                                containing the mode shapes and the other
c                                records containing zeros, where N is the
c                                number of modes actually used in the harmonic
c                                analysis. Order corresponds to the DOF order
c                                given in record at ptrDOF.

c                                If the analysis uses the mode superposition
c                                method (using the subspace mode extraction
c                                technique), this record will not be present.
c                                  (psi(i,j),i=1,nmrow) (or ac)

c  FRQ      dp      1     nmrow   Frequencies extracted from the modal analysis.
c                                This record is present only for analyses using
c                                the mode superposition method (using the
c                                reduced mode extraction technique).
c                                 (freq(i),i=1,nmrow)

c  DSP      cmp  ncumit  nmrow+5  Calculated complex displacements
c                                The first nmrow entries are the displacements
c                                in the same order as the original set of DOFs
c                                (see record AFTER ptrDOF). For the last five
c                                entries:
c                                   Real part             Imag part
c                                1. frequency for these   frequency increment
c                                   values
c                                2. load step number      substep number
c                                3. cumulative iteration  zero
c                                   number
c                                4. zero                  zero
c                                5. scale factor          zero
c                                   (zero if the
c                                   analysis uses the
c                                   reduced method)
c                                     (cvs(i),i=1,nmrow),(freq,delf),
c                                     (itime,itter),(ncumit,0.0),(0.0,0.0),
c                                     (fscale,0.0)
c                                Note: If, upon reading of this record, there
c                                is less than nmrow+5 items in the record,
c                                then only a selected set of nodes were
c                                output. Another record follows (integer, less
c                                than lenbac long) which contains the list of
c                                nodes for which DOF solutions are available.
```

1.5. Description of the Modal Results File

This section explains the content of the modal results file (**jobname.mode**).

1.5.1. Standard ANSYS File Header

See Section 1.1.2: The Standard Header for ANSYS Binary Files for a description of this set. File number (Item 1) is 9.

1.5.2. MODE File Format

```
*comdeck,fdmode
c *** ansys(r) copyright(c) 2000
```

```
c *** ansys, inc.

c     **********  description of modal result file  **********

c *** mpg fdmode < modspc romstr lire_freq_mode lire_nb_mode: mode file desc

      character*8  MODENM
      parameter  (MODENM='mode    ')
      LONGINT         modefpL, modefp
      integer         modebk,  modeut

      common /fdmode/ modefpL, modebk, modeut
      equivalence  (modefp,modefpL)

c   ********** common variable descriptions ***********
co  modefpL      file position on file mode
co  modebk       block number for file mode
co  modeut       file unit for file mode


c   See fddesc for documentation of how binary files are stored.
c
c     **********  file format  **********

c        recid tells the identifier for this record.  Not all records will have
c             identifiers -- they are only indicated for those records whose
c             record pointers are stored in the second file header.

c        type tells what kind of information is stored in this record:
c             i - integer
c             dp - double precision
c             cmp - complex

c        nrec tells how many records of this description are found here

c        lrec tells how long the records are (how many items are stored)

c recid    type    nrec    lrec     contents

c  ---      i       1      100      standard ANSYS file header (see binhed for
c                                   details of header contents)


c  ---      i       1       60      .MODE FILE HEADER
c
c                                     fun09,  nmrow, nmatrx,  nmode, numdof,
c                                      maxn,  wfmax, lenbac,      0,  nontp,
c                                    lumpms, extopt, SvCode,    kan, ldstep,
c                                    numitr, expbeg, expend, nspect, nSPdat,
c                                    ptrRDF, ptrFRQ, ptrPRT, ptrSHP, ptrLOD,
c                                    ptrSTF, ptrMAS, ptrDMP, ptrCOF, ptrDCF,
c                                    ptrLPM, ptrSP1,ptrSHPh,ptrLODh,ptrSTFh,
c                                  ptrMASh,ptrDMPh,ptrLPMh,ptrSP1h,ptrIRHSl,
c                                 ptrIRHSh,PowerDyn,     0,      0,      0,
c                                        0,      0,      0,      0,      0,
c                                        0,      0,      0,      0,      0,
c                                        0,      0,      0,      0,      0
c                                  each item in header is described below:

c                                    fun09  - unit number (mode file is 9)
c                                    nmrow  - number of rows/columns in matrices
c                                             (maxn*numdof). If extopt = 0, nmrow
c                                             is the number of rows in the
c                                             reduced matrices and the number of
c                                             master degrees of freedom.
c                                    nmatrx - number of reduced matrices on the
c                                             file (applies only if extopt=0)
c                                    nmode  - number of modes extracted
c                                    numdof - number of dof per node
c                                    maxn   - maximum node number (If extopt = 3
c                                             or 4, the actual number of nodes is
c                                             referenced.)
```

```
c                                  wfmax  - maximum wavefront (Does not apply
c                                           if extopt = 3 or 4.)
c                                  lenbac - number of nodes
c                                  0      - position not used
c                                  nontp  - number of equations on the .TRI
c                                           file (Does not apply if extopt =
c                                           0.)
c                                  lumpms - lumped mass key
c                                           = 0 - default matrix type
c                                           = 1 - lumped
c                                           (Does not apply if extopt = 3 or
c                                           4.)
c                                  extopt - mode extraction method
c                                           = 0 - reduced
c                                           = 1 - subspace
c                                           = 3 - unsymmetric Lanczos
c                                           = 4 - damped Lanczos
c                                           = 6 - block Lanczos
c                                           = 7 - QR damped
c                                           = 8 - AMLS
c                                  SvCode - Solver assembly code
c                                           = 0 Frontal assembly (SV_ANSYS)
c                                           = 1 Symbolic assembly (SV_CASI)
c                                  kan    - analysis type
c                                           = 1 - buckling
c                                           = 2 - modal
c                                  ldstep - load step number
c                                  numitr - total number of cumulative
c                                           iterations done during analysis
c                                           (Does not apply if extopt = 3 or
c                                           4.)
c                                  expbeg - beginning of the frequency range of
c                                           interest
c                                  expend - end of the frequency range of
c                                           interest
c                                  nspect - number of spectra
c                                  nSPdat - number of data items per spectrum
c                                  ptrRDF - pointer to reduced degree of
c                                           freedom set used in model
c                                  ptrFRQ - pointer to the frequencies
c                                  ptrPRT - pointer to the participation
c                                           factors
c                                  ptrSHP - pointer to the mode shapes
c                                           (eigenvectors)
c                                  ptrLOD - pointer to the load vectors
c                                  ptrSTF - pointer to the reduced stiffness
c                                           matrix
c                                  ptrMAS - pointer to the reduced mass matrix
c                                  ptrDMP - pointer to the reduced damping
c                                           matrix
c                                           (if extopt=7 : pointer to the modal
c                                             damping matrix)
c                                  ptrCOF - pointer to the mode coefficients
c                                  ptrDCF - pointer to the modal damping
c                                           coefficients
c                                  ptrLPM - pointer to the diagonal mass vector
c                                  ptrSP1 - pointer to the the spectrum data
c                               ptrIRHSl,h - pointer to imaginary part of RHS vector
c                                 PowerDyn - PowerDynamics key (currently only set in subout.F)
c                                           = 0 Subspace, Block Lanczos, etc... methods
c                                           = 1 Powerdynamics method
c                                  0      - position not used
c                                  0      - position not used
c                                  0      - position not used
c                                  0      - position not used
c                                  0      - position not used
c                                  0      - position not used

c  ---      i       1     numdof    Degrees of freedom per node
c                                   DOF reference numbers are:
c         UX  = 1, UY  = 2, UZ  = 3, ROTX= 4, ROTY= 5, ROTZ= 6, AX  = 7, AY  = 8
c         AZ  = 9, VX  =10, VY  =11, VZ  =12  ****** 13-18 are spares **********
```

```
c         ***************  PRES=19, TEMP=20, VOLT=21, MAG =22, ENKE=23, ENDS=24
c         EMF =25, CURR=26  ********* 27-32 are spares ************************
c                                    (curdof(i),i=1,numdof)

c  ---      i       1     lenbac    Nodal equivalence table
c                                    This table equates the number used for
c                                    storage to the actual node number.
c                                    (baclst(i),i=1,lenbac)

c  RDF      i       1     nmrow     Reduced set of degrees of freedom used.
c                                    This record is present only if extopt = 0
c                                    The DOFs are calculated as (N-1)*NUMDOF+DOF,
c                                    where N is position number of the node in
c                                    the nodal equivalence table and DOF is the
c                                    DOF reference number given above
c                                    (l(i),i=1,nmrow) (if nmatrx>0)

c  FRQ      dp      1     nmode     Frequencies (eigenvalues). Frequencies are
c                                    complex if extopt=3 or 4. Numbers stored are
c                                    the squares of the natural circular
c                                    frequencies (w**2, where w=radians/time).
c                                    You can obtain the natural frequencies, f
c                                    (in cycles/time), using the equation f=w/2pi
c                                    (freq(i),i=1,nmode)

c  PRT      dp      1     nmode     Participation factors. Factors are complex
c                                    if EXTOPT=3 or 4.
c                                    (pfact(i),i=1,nmode)

c  COF      dp      1     nmode     Mode coefficients
c                                    (modecf(i),i=1,nmode)

c  DCF      dp      1     nmode     Modal damping coefficients
c                                    (dampmd(i),i=1,nmode)

c  SHP      dp    nmode   nmrow     Mode shapes (eigenvectors). Mode shapes are
c                                    complex if extopt=3 or 4. If extopt=0, the
c                                    mode shape order corresponds to the DOF list
c                                    stored at position ptrRDF.  If extopt does
c                                    not equal 0, the order corresponds to the
c                                    nodal equivalence table
c                                    (psi(i,j),i=1,nmrow)

c  LOD      dp      1     nmrow     Load vector. This record is present only if
c                                    extopt=0 or 1.
c                                    (f(i),i=1,nmrow)

c IRHS      dp      1     nmrow     Imaginary Load vector.  This record is present
c                                    only if extopt = 6.

c  LPM      dp      1     nmrow     Lumped mass vector. This record is present
c                                    only if lumpms=1 and nmatrix=0. It is a
c                                    vector containing the mass at each node in
c                                    the system.
c                                    (mass(i),i=1,nmrow)

c  STF      dp    nmrow   nmrow     Reduced stiffness matrix. Each row of the
c                                    matrix is stored as a record. The matrix is
c                                    present only if nmatrx > 0. Row order is the
c                                    same as the DOF order stored at position
c                                    ptrRDF.
c                                    (ak(i,j),i=1,nmrow)

c  MAS      dp    nmrow   nmrow     Reduced mass matrix. Each row of the matrix
c                                    is stored as a record. The matrix is present
c                                    only if nmatrx > 1. Row order is the same as
c                                    the DOF order stored at position ptrRDF.
c                                    (am(i,j),i=1,nmrow)

c  DMP      dp    nmrow   nmrow     Reduced damping matrix. Each row of the
c                                    matrix is stored as a record. The matrix is
c                                    present only if nmatrx > 2. Row order is the
```

```
c                                  same as the DOF order stored at position
c                                  ptrRDF.
c                                   (ac(i,j),i=1,nmrow)

c for each spectrum (nspect records):
c  SP1      dp      1     nmode     Mode coeff for this spectra
c  ---      dp      1     nmode     Modal damping values
c  ---      dp      1     130       svcom: freqtb, etc.
c  ---      dp      1     20        misc. spectra data
```

1.6. Description of the Element Matrices File

This section explains the content of the element matrices file (**jobname.emat**).

1.6.1. Standard ANSYS File Header

See Section 1.1.2: The Standard Header for ANSYS Binary Files for a description of this set. File number (Item 1) is 2.

1.6.2. EMAT File Format

```
*comdeck,fdemat
c *** ansys(r) copyright(c) 2000
c *** ansys, inc.

c     **********  description of element matrix file  **********
c
c *** mpg fdemat.inc < eoelem elostr eofini outelm elfini EmatAssemble sffini
c          eqprep sfform elstrt slvstr: emat file description
c
      character*8  EMATNM
      parameter  (EMATNM='emat    ')

      LONGINT         ematfpL, ematfp
      integer         ematbk, ematut
      common /fdemat/ ematfpL, ematbk, ematut
      equivalence  (ematfp,ematfpL)

c   ********** common variable descriptions ***********
co  ematfpL      file position on file emat
co  ematbk       block number for file emat
co  ematut       file unit for file emat

c   See fddesc for documentation of how binary files are stored.
c
c     **********  file format  **********

c        recid tells the identifier for this record.  Not all records will have
c             identifiers -- they are only indicated for those records whose
c             record pointers are stored in the second file header.

c        type tells what kind of information is stored in this record:
c             i - integer
c             dp - double precision
c             cmp - complex

c        nrec tells how many records of this description are found here

c        lrec tells how long the records are (how many items are stored)

c recid    type    nrec    lrec     contents

c  ---      i       1      100      standard ANSYS file header (see binhed for
c                                   details of header contents)

c  ---      i       1       40      .EMAT FILE HEADER
c
```

```
c                                fun02,   nume, numdof,   lenu, lenbac,
c                                 maxn,      0,      0, nodref,  lumpm,
c                                kygst,   kygm,   kycd,  kygss,  kygaf,
c                                kygrf,      0,      0,      0,      0,
c                              ptrElmh,ptrFSTh,ptrLSTh,ptrBITh,ptrEHDh,
c                              ptrIDXh,  numCE,maxLeng, ptrCEl, ptrCEh,
c                               ptrDOF, ptrBAC,ptrELMl,ptrFSTl,ptrLSTl,
c                              ptrBITl,ptrEHDl,ptrIDXl,ptrendH,ptrendL

c                             each item in header is described below:

c                                fun02  - unit number (emat file is 2)
c                                nume   - number of elements
c                                numdof - number of dofs per node
c                                lenu   - total DOFs of model
c                                lenbac - number of nodes
c                                maxn   - maximum node number
c                                0      - position not used
c                                0      - position not used
c                                nodref - actual number of nodes referenced
c                                lumpm  - lumped mass key
c                                         = 0 - default matrix type
c                                         = 1 - lumped
c                                kygst  - global stiffness matrix calculate
c                                         key
c                                         = 0 - do not calculate
c                                         = 1 - calculate
c                                kygm   - global mass matrix calculate key
c                                         = 0 - do not calculate
c                                         = 1 - calculate
c                                kycd   - global damping matrix calculate key
c                                         = 0 - do not calculate
c                                         = 1 - calculate
c                                kygss  - global stress stiffening matrix
c                                         calculate key
c                                         = 0 - do not calculate
c                                         = 1 - calculate
c                                kygaf  - global applied force vector
c                                         calculate key
c                                         = 0 - do not calculate
c                                         = 1 - calculate
c                                kygrf  - global restoring force vector
c                                         calculate key (Newton-Raphson only)
c                                         = 0 - do not calculate
c                                         = 1 - calculate
c                                0      - position not used
c                                0      - position not used
c                                0      - position not used
c                                0      - position not used

c                                ptrELMh- Highpointer to element equivalence
c                                         table
c                                ptrFSTh- High pointer to first element at a
c                                         DOF table
c                                ptrLSTh- High pointer to last element at a
c                                         DOF table
c                                ptrBITh- High pointer to dof bits
c                                ptrEHDh- High pointer to the start of the
c                                         element matrices
c                                ptrIDXh- High pointer to element matrices
c                                         index table

c                                numCE  - number of internal CEs
c                                maxLeng- maximum length of any internal CE
c                                ptrCEl - low pointer to internal CE list
c                                ptrCEh - high pointer to internal CE list
c                                ptrDOF - pointer to degrees of freedom per
c                                         node used in model
c                                ptrBAC - pointer to nodal equivalence table

c                                ptrELMl- Low pointer to element equivalence
c                                         table
```

```
c                                  ptrFSTl- Low pointer to first element at a
c                                           DOF table
c                                  ptrLSTl- Low pointer to last element at a
c                                           DOF table
c                                  ptrBITl- Low pointer to dof bits
c                                  ptrEHDl- Low pointer to the start of the
c                                           element matrices
c                                  ptrIDXl- Low pointer to element matrices
c                                           index table

c                                  ptrendH- High pointer to end of file
c                                  ptrendL- Low  pointer to end of file

c       Note: the analysis type sets the global calculate keys.

c  ---     dp       1       20      Time information
c
c                                  timval, timinc, frqval, timbeg, timend,
c                                     0.0,    0.0,    0.0,    0.0,    0.0,
c                                     0.0,    0.0,    0.0,    0.0,    0.0,
c                                     0.0,    0.0,    0.0,    0.0,    0.0,

c                                 each item is described below:

c                                  timval - the current time
c                                  timinc - the time increment
c                                  frqval - the current frequency (from a
c                                           harmonic analysis)
c                                  timbeg - the start time for the analysis
c                                  timend - the end time for the analysis
c                                  0.0    - position not used
c                                  0.0    - position not used
c                                  0.0    - position not used
c                                  0.0    - position not used
c                                  0.0    - position not used
c                                  0.0    - position not used
c                                  0.0    - position not used
c                                  0.0    - position not used
c                                  0.0    - position not used
c                                  0.0    - position not used
c                                  0.0    - position not used
c                                  0.0    - position not used
c                                  0.0    - position not used
c                                  0.0    - position not used

c  DOF      i       1    numdof     Degrees of freedom per node
c                                  DOF reference numbers are:
c        UX  = 1, UY  = 2, UZ  = 3, ROTX= 4, ROTY= 5, ROTZ= 6, AX  = 7, AY  = 8
c        AZ  = 9, VX  =10, VY  =11, VZ  =12  ****** 13-18 are spares **********
c        ****************  PRES=19, TEMP=20, VOLT=21, MAG =22, ENKE=23, ENDS=24
c        EMF =25, CURR=26  ********* 27-32 are spares ************************
c                                   (curdof(i),i=1,numdof)

c  BAC      i       1    lenbac     Nodal equivalence table. This table equates
c                                  the number used for storage to the actual
c                                  node number
c                                   (baclst(i),i=1,lenbac)

c  ELM      i       1     nume      Element equivalence table. The ANSYS program
c                                  stores all element data in the numerical
c                                  order that the SOLUTION processor solves the
c                                  elements.  This table equates the order
c                                  number used to the actual element number
c                                   (eorder(i),i=1,nume)

c  FST      i       1     lenu      First element at a DOF table. This record
c                                  signifies the first element encountered at a
c                                  particular DOF.
c                                   (frstel(i),i=1,lenu)

c  LST      i       1     lenu      Last element at a DOF table. This record
```

```
c                                  signifies the last element encountered at a
c                                  particular DOF.
c                                   (lastel(i),i=1,lenu)

c  BIT      i       1     lenu      Bits set at a DOF table. This record
c                                  has bits for constraints, forces, etc.
c                                   (DofBits(i),i=1,lenu) (added at 10.0)

c  IDX      i       1     2*nume    Element index table. This record specifies
c                                  the file location for the beginning of the
c                                  data for each element.
c                                   (index(i),i=1,nume) Low part of pointer
c                                   (index(i),i=1,nume) High part of pointer


c  The records at the end of the file store element information and get written
c  as a set for each element(nume sets of these records will appear on the file
c  at this point) ptrEHD indicates the beginning of the element data.

c  If substructure matrices are written to the EMAT file, they are written in a
c  different format than is shown here. This alternate format is not documented
c  at this time, as it is likely to change in the future.


c  EHD      i       1      10       Element matrix header
c
c                                   stkey,   mkey,   dkey,  sskey, akey,
c                                   nrkey,   ikey,      0,      0, nmrow

c                                  each item in header is described below:

c                                   stkey  - stiffness matrix key
c                                            = 0 - matrix not present
c                                            = 1 - matrix present
c                                   mkey   - mass matrix key
c                                            = 0 - matirx not present
c                                            = 1 - matrix present
c                                   dkey   - damping matrix key
c                                            = 0 - matrix not present
c                                            = 1 - matrix present
c                                   sskey  - stress stiffening matrix key
c                                            = 0 - matrix not present
c                                            = 1 - matrix present
c                                   akey   - applied load vector key
c                                            = 0 - vector not used
c                                            = 1 - vector used
c                                   nrkey  - newton-raphson(restoring) load
c                                            vector key (for nonlinear analyses)
c                                            = 0 - vector not used
c                                            = 1 - vector used
c                                   ikey   - imaginary load vector key
c                                              (for complex analyses)
c                                            = 0 - vector not used
c                                            = 1 - vector used
c                                   0      - position not used
c                                   0      - position not used
c                                   nmrow  - numbers/columns in matrices. If the
c                                            number is negative, the matrices
c                                            will be written in lower triangular
c                                            form.

c  ---      i       1    nmrow      DOF index table. This record specifies the
c                                  DOF locations of this element matrix in
c                                  relation to the global matrix. The index is
c                                  calculated as (N-1)*NUMDOF+DOF, where N is
c                                  the position number of the node in the nodal
c                                  equivalence table and DOF is the DOF
c                                  reference number given above

c  ---     dp    varies  varies     Element matrices. This record is repeated
c                                  for each stiffness, mass, damping, and
c                                  stress stiffening matrix. If the matrix is
```

```
c                                  diagonal, the length of the records will be
c                                  nmrow.  If the matrix is unsymmetric, the
c                                  length of the records will be nmrow*nmrow.
c                                  If the matrix is symmetric, only the lower
c                                  triangular terms are written and the length
c                                  of the records will be (nmrow)*(nmrow+1)/2.

c  ---     dp       1    2*nmrow    Element force vectors. This record contains
c                                  both the applied force vector and the
c                                  (restoring or imaginary) load vector.
c
c
c      *************** Internal CE information ***********************
c      The following records repeat numCE times... one for each internal
c      CE created during solution... these are stored here for the psolve
c      command, such as the case of a prestressed nonlinear modal analysis
c
c  CE      i       3     numCE      First part is the CE number, the second part is
c                                  the number of terms in this internal CE, and
c                                  the third part is the external element number
c                                  of the element that created this internal CE
c
c  ---     i     nTerms  numCE      integer info (list of node*32 + dof)
c
c  ---    dp     nTerms  numCE      dp info (list of coefficients including constant term)
c
c
c   kygst        global stiffness matrix calculate key
c   kygm         global mass matrix calculate key
c   kygd         global damping matrix calculate key
c   kygss        global stress stiffening matrix calculate key
c   kygaf        global applied force matrix calculate key
c   kygrf        global restoring force matrix calculate key
```

1.7. Description of the Substructure Matrices File

This section explains the contents of the substructure matrices file (**jobname.sub**).

1.7.1. Standard ANSYS File Header

See Section 1.1.2: The Standard Header for ANSYS Binary Files for a description of this set. File number (Item 1) is 8.

1.7.2. SUB File Format

```
*comdeck,fdsub
c *** ansys(r) copyright(c) 2000
c *** ansys, inc


c     **********  description of substructure matrix file  **********
      character*8  SUBNM
      parameter  (SUBNM='sub    ')
      LONGINT        subfpL, lenSubL, subfp
      integer        subbk, subut
      common /fdsub/ subfpL, lenSubL, subbk, subut
      equivalence  (subfp,subfpL)

c  write:  matout
c  read:

c   ********** common variable descriptions ***********
co  subfp       file position on file sub
co  subbk       block number for file sub
co  subut       file unit for file sub
co  lenSub      length of sub file (saved for slvdta.F)
```

```
c   See fddesc for documentation of how binary files are stored.
c
c     **********  file format  **********

c     recid tells the identifier for this record.  Not all records will have
c          identifiers -- they are only indicated for those records whose
c          record pointers are stored in the second file header.

c     type tells what kind of information is stored in this record:
c          i - integer
c          dp - double precision
c          cmp - complex

c     nrec tells how many records of this description are found here

c     lrec tells how long the records are (how many items are stored)

c recid    type    nrec    lrec     contents

c  ---      i       1      100      standard ANSYS file header (see binhed
c                                   for details of header contents)

c  HED      i       1       60      .SUB FILE HEADER  (FULL MATRICES)
c
c                                  fun08,  nmrow, nmatrx,  nedge, numdof,
c                                   maxn,  wfmax, lenbac,   nnod, kunsym,
c                                   kstf,  kmass,  kdamp,    kss,  nvect,
c                                 nWorkL,  lenU1, sesort, lenlst,ptrLodL,
c                                 ntrans, ptrMtx, ptrXFM, ptrHED,  name1,
c                                  name2,   lrok,   trok,  name3,  name4,
c                                 ptrDOF, ptrDST, ptrBAC, ptrTIT, ptrNOD,
c                                 ptrXYZ, ptrEDG, ptrGDF, thsubs, ptrPOS,
c                                 ptrORG, stfmax,ptrLodH, nmodes, keydim,
c                               cmsMethod,  name5,  name6,  name7,  name8,
c                                 nvnodes,ptrCXFM, nWorkh,      0,      0,
c                                       0,      0,      0,      0,      0

c  HED      i       1       60      .SUB FILE HEADER  (SPARSE MATRICES)
c
c                                  fun08,  nmrow, nmatrx,       , numdof,
c                                   maxn,       , lenbac,   nnod, kunsym,
c                                   kstf,  kmass,  kdamp,       ,  nvect,
c                                 nTermL,       ,       ,       ,ptrLodL,
c                                       ,ptrMtxL,       , ptrHED,  name1,
c                                  name2,       ,       ,  name3,  name4,
c                                 ptrDOF,       , ptrBAC, ptrTIT, ptrNOD,
c                                 ptrXYZ,       ,       , thsubs,       ,
c                                       , stfmax,ptrLodH,       ,       ,
c                                       ,  name5,  name6,  name7,  name8,
c                                       ,       , nTermH,ptrMtxH,ptrColL,
c                               ptrColH,ptrCofL,ptrCofH,      0,      0

c                                 each item in header is described below:

c                                  fun08  - unit number (sub file is 8)
c                                  nmrow  - number of rows in matrices (also
c                                           number of dofs in substructure)
c                                  nmatrx - number of matrices on file
c                                  nedge  - number of edges for outline
c                                  numdof - number of dofs per node
c                                  maxn   - maximum node number of complete
c                                           model presently in database
c                                  wfmax  - maximum wavefront of substruct.
c                                           during generation pass
c                                  lenbac - number of nodes defining
c                                           substructure during the
c                                           generation pass
c                                  nnod   - number of unique nodes in the
c                                           substructure having DOFs, and
c                                           which define this substructure
c                                           during the use pass.  Also, the
c                                           number of nodes having master
```

```
c                                  DOFs.
c                              kunsym - unsymmetric matrix key
c                                  = 0 - symmetric
c                                  = 1 - unsymmetric
c                              kstf   - stiffness matrix present key
c                                  = 0 - matrix is not on file
c                                  = 1 - matrix is on file
c                              kmass  - mass matrix present key
c                                  = 0 - matrix is not on file
c                                  = 1 - matrix is on file
c                                  =-1 - Lumped mass vestor (Sparse only)
c                              kdamp  - damping matrix present key
c                                  = 0 - matrix is not on file
c                                  = 1 - matrix is on file
c                              kss    - stress stiffening matrx present
c                                  = 0 - matrix is not on file
c                                  = 1 - matrix is on file
c                              nvect  - number of load vectors
c                                  (at least 1 is required)
c                              nWorkL,H  - BCS workspace length (only for
c                                  bacsub)
c                              nTermL,H  - Number of terms in sparse matrix
c                              lenU1  - length of intermediate transformation
c                                  vector
c                              sesort - DOF set sort key
c                                  = 0 - numbers are not sorted
c                                  = 1 - numbers are sorted in
c                                        ascending order
c                              lenlst - maximum length of DOF set for
c                                  this substructure (maxn*numdof)
c                              ptrLod - pointer to the start of the load
c                                  vectors (see also ptrLodh)
c                              ntrans - transformed key
c                                  = 0 - substructure has not been
c                                        transformed
c                                  > 0 - substructure copied
c                                        from another substructure,
c                                        via either SESSYM or SETRAN
c                              ptrMtxL,H - pointer to the start of the
c                                  substructure matrices (iDiagL for
c                                  sparse matrices)
c                              ptrXFM - pointer to the substructure
c                                  transformations
c                              ptrHED - pointer to the SUB file header
c                              name1  - first four characters of the
c                                  substructure file name, in
c                                  integer form
c                              name2  - second four characters of the
c                                  substructure file name, in
c                                  integer form
c                              lrok   - large rotation key
c                                  = 0 - DOF set is not valid for
c                                        large rotations
c                                  = 1 - DOF set is valid
c                              trok   - SETRAN rotation key
c                                  = 0 - DOF set is not valid for
c                                        SETRAN nodal rotation
c                                  = 1 - DOF set is valid
c                              name3  - third four characters of the
c                                  substructure file name, in
c                                  integer form
c                              name4  - fourth four characters of the
c                                  substructure file name, in
c                                  integer form
c                              ptrDOF - pointer to the DOF/node list
c                              ptrDST - pointer to the local DOF set
c                              ptrBAC - pointer to the nodes comprising
c                                  the substructure
c                              ptrTIT - pointer to the title
c                              ptrNOD - pointer to the unique nodes
c                                  defining the substructure
c                              ptrXYZ - pointer to the coordinates of the
```

```
c                                       unique nodes
c                                  ptrEDG - pointer to the substructure edges
c                                  ptrGDF - pointer to the global DOF set
c                                  thsubs - thermal key
c                                           = 0 - structural
c                                           = 1 - thermal
c                                  ptrPOS - pointer to the sorted substructure
c                                           DOF set to the original
c                                  ptrORG - pointer to the DOF set of the model
c                                           during the generation pass
c                                  stfmax - maximum diagonal stiffness term
c                                           (packed into an integer)
c                                  ptrLodh- High 32 bits of 64 bit pointer
c                                  nmodes - number of modes used to generate
c                                           CMS s.e.
c                                  keydim - dimensionality key
c                                           = 1 - axisymmetric
c                                           = 2 - 2-D
c                                           = 3 - 3-D
c                                cmsMethod - component mode synthesis method
c                                  name5  - fifth four characters of the
c                                           substructure file name, in integer
c                                           form
c                                  name6  - sixth four characters of the
c                                           substructure file name, in integer
c                                           form
c                                  name7  - seventh four characters of the
c                                           substructure file name, in integer
c                                           form
c                                  name8  - eighth four characters of the
c                                           substructure file name, in integer
c                                           form
c                                 nvnodes - number of virtual nodes that contain
c                                           the modal coordinates
c                                 ptrCXTM - coordinate transformation
c                                 ptrColL,H - pointer to the iCol sparse matrix
c                                             array
c                                 ptrCofL,H - pointer to the of the
c                                             sparse matrix Sk(1:nTerm),
c                                             Sm(1:nTermL),Sc(1:nTermL),
c                                             Ss(1:nTermL)  Each matrix is a
c                                             single large record

c                                  note: name1/2/3/4/5/6/7/8 are the
c                                        inexc4 representation of the
c                                        32 character filename.
c                                        name1/2/5/6/7/8 will be "0"
c                                        for pre rev 5.2 files - cwa

c   Note: If ntrans > 0, records from position ptrDOF to ptrGDF will be
c         identical to the data for the copied substructure.


c  XFM      dp      1      125   Substructure transformations (5*25 double
c                                precisions). This record has meaning only
c                                if ntrans > 0. You can define up to five
c                                levels of transformations, with 25 variables
c                                in each level.  Up to the first seven
c                                variables are used as follows:
c
c                                If the substructure was transferred (via the
c                                 SETRAN command):
c                                1st variable - 1.0
c                                2nd variable - nodal increment
c                                3rd variable - reference number of
c                                 coordinate system where substructure will
c                                 be transferred
c                                4th variable - reference number of
c                                 coordinate system where substructure is
c                                 presently defined
c                                5th variable - x coordinate increment
c                                6th variable - y coordinate increment
```

```
c                                7th variable - z coordinate increment

c                                If the substructure used symmetry (via the
c                                 SESYMM command):
c                                1st variable - 2.0
c                                2nd variable - nodal increment
c                                3rd variable - number of coordinate
c                                 component to be used in operation
c                                 = 1 - x coordinate
c                                 = 2 - y coordinate
c                                 = 3 - z coordinate
c                                4th variable - reference number of
c                                 coordinate system to be used for symmetry
c                                 operation
c  XFM      dp      1      250   Substructure transformations (5*25 double

c  DOF      i       1     numdof    Degrees of freedom per node  (Global)
c                                     (curdof(i),i=1,numdof)
c                                    DOF reference numbers are:
c    UX  = 1, UY  = 2, UZ  = 3, ROTX= 4, ROTY= 5, ROTZ= 6, AX  = 7, AY  = 8
c    AZ  = 9, VX  =10, VY  =11, VZ  =12  ****** 13-18 are spares **********
c    ***************  PRES=19, TEMP=20, VOLT=21, MAG =22, ENKE=23, ENDS=24
c    EMF =25, CURR=26  ********* 27-32 are spares ************************

c  DST      i       1     nmrow  This record contains degrees of freedom for
c                                this substructure of the unique nodes, as
c                                used with this substructure, in ascending
c                                order. This index is calculated as
c                                (N-1)*numdof+DOF, where N is the node number
c                                and DOF is the DOF reference number given
c                                above
c                                 (lsort(i),i=1,nmrow)

c  POS      i       1     nmrow  This record stores the positions of the
c                                local DOF set in relation to the generated
c                                DOF set. (lposit(i),i=1,nmrow)

c  ORG      i       1     nmrow  DOF set of the model as defined during the
c                                generation pass. This index is calculated as
c                                (N-1)*NUMDOF+DOF, where N is the position
c                                number of the node in the nodal equivalence
c                                table and DOF is the DOF reference number
c                                given above
c                                 (lorig(i),i=1,nmrow)

c  BAC      i       1     lenbac This group describes nodes that defined the
c                                substructure during the generation pass of
c                                the analysis. Nodal data is stored in arrays
c                                equal to the number of used or referenced
c                                nodes. This table equates the number used
c                                for storage to the actual node number.
c                                 (baclst(i),i=1,lenbac)

c  TIT      i       1       20   Substructure title (converted to integers -
c                                see inexc4)

c  NOD      i       1      nnod  This record describes unique nodes defining
c                                the substructure for the use pass of the
c                                analysis.  These are also the nodes having
c                                master degrees of freedom.
c                                 (node(i),i=1,nnod)

c  XYZ     dp     nnod      6    This record describes the coordinates of a
c                                unique node, in the order X, Y, Z, THXY,
c                                THYZ, and THZX.  Nodal order corresponds to
c                                that of the node list given above
c                                 (xyzang(j,i),j=1,6)

c  EDG     dp     nedge     6    This record contains beginning and ending
c                                locations (X1,Y1,Z1,X2,Y2,Z2 coordinates) of
c                                a straight line comprising an edge of the
c                                substructure.
```

```
c  GDF      i       1     nmrow  This record describes global degrees of
c                               freedom of the unique nodes in ascending
c                               order, as used during the analysis use pass.
c                               This index is calculated as (N-1)*32+DOF,
c                               where N is the node number and DOF is the
c                               DOF reference number given above
c                                (l(i),i=1,nmrow) (sorted)


c The substructure matrices are written at this position in the file.  One row
c  of each matrix is written to the file at a time. i.e. the first row of each
c  matrix is written, then the second row of each matrix, etc. this pattern
c  continues until all nmrow rows of each matrix have been written to the file.

c  MAT      dp      1     nmrow   Row of the stiffness matrix, if nmatrx > 0.
c                                  (ak(i,j),i=1,nmrow)
c  ---      dp      1     nmrow   Row of the mass matrix, if nmatrx > 1.
c                                  (am(i,j),i=1,nmrow)
c  ---      dp      1     nmrow   Row of the damping matrix, if nmatrx > 2.
c                                  (ac(i,j),i=1,nmrow)
c  ---      dp      1     nmrow   Row of the stress stiffening matrix, if
c                                  nmatrx > 3.
c                                  (gs(i,j),i=1,nmrow)

c  LOD      dp   nvect    nmrow   This record contains the load vectors.
c                                  (f(i),i=1,nmrow)
```

1.8. Description of the Triangularized Stiffness File

This section explains the contents of the triangularized stiffness file (***jobname*.tri**).

1.8.1. Standard ANSYS File Header

See Section 1.1.2: The Standard Header for ANSYS Binary Files for a description of this set. File number (Item 1) is 11.

1.8.2. TRI File Format

```
*comdeck,fdtri
c *** ansys(r) copyright(c) 2000
c *** ansys, inc
c
c
c     **********  description of triangularized stiffness file  **********
c
c *** mpg fdtri.inc < stff10 slvstr romstr modstr: tri file description
c
      character*8  trinm, TriName
      parameter  (trinm='tri     ',TriName='Tri Bufr')

      LONGINT  trifp
      integer  tribk,  triut
      common /fdtri/   trifp,  tribk,  triut

c  open:     slvstr                     subspc
c  write:    eqsol,eqclos,lfout,cfout   subtri
c  read:     stff10,bacfil              subfwd,subbac
c  close:    bacfil,svkan2              subspc
c
c   ********** common variable descriptions ***********
co  trifp       file position on file tri  (LONGINT)
co  tribk       block number for file tri
co  triut       file unit for file tri
c
c   See fddesc for documentation of how binary files are stored.
c
```

```
c   The TRI file is generated for static and reduced[modal,harmonic,transient,
c    substructure] analyses
c
c     **********  file format  **********
c
c        recid tells the identifier for this record.  Not all records will have
c             identifiers -- they are only indicated for those records whose
c             record pointers are stored in the second file header.
c
c        type tells what kind of information is stored in this record:
c             i - integer
c             dp - double precision
c             cmp - complex
c
c        nrec tells how many records of this description are found here
c
c        lrec tells how long the records are (how many items are stored)
c
c recid    type    nrec    lrec     contents
c
c  ---      i       1      100      standard ANSYS file header (see binhed for
c                                   details of header contents)
c
c  ---      i       1       20      .TRI FILE HEADER
c
c                                    fun11,  nontp,  nmast,      1,    kan,
c                                    wfmax, lenbac, numdof, ptrMST, ptrend,
c                                    lumpm, keyuns, ptrMS1, ptrMS2, ptrEN1,
c                                   ptrEN2,      0,      0, ptrTRI,      0
c
c                                  each item in header is described below:
c
c                                    fun11  - unit number (tri file is 11)
c                                    nontp  - number of equations on file
c                                    nmast  - number of master dofs
c                                    1      - position not used, always = 1
c                                    kan    - analysis type
c                                    wfmax  - maximum wavefront
c                                    lenbac - number of nodes
c                                    numdof - number of degrees of freedom (DOF)
c                                             per node
c                                    ptrMST - 32 bit pointer to the master dof
c                                             list, only here for backward
c                                             compatibility.  Do not use if
c                                             ptrMS1 or ptrMS2 are non-zero
c                                    ptrend - 32 bit pointer to the end of file
c                                             only here for backward
c                                             compatibility.  Do not use if
c                                             ptrEN1 or ptrEN2 are non-zero
c                                    lumpm  - lumped mass key
c                                             = 0 - default matrix type
c                                             = 1 - lumped
c                                    keyuns - unsymmetric key
c                                             = 0 - the matrix is not unsymmetric
c                                             = 1 - the matrix is unsymmetric
c                                    ptrMS1,
c                                    ptrMS2 - These two values are two halves of
c                                             a 64 bit pointer that points to the
c                                             master dof list
c                                    ptrEN1,
c                                    ptrEN2 - These two values are two halves of
c                                             a 64 bit pointer that points to the
c                                             end of file
c                                    0      - position not used
c                                    0      - position not used
c                                    ptrTRI - pointer to the beginning of the
c                                             triangularized matrix data
c                                    0      - position not used
c
c  ---      i       1      numdof   Degrees of freedom per node
c                                   DOF reference numbers are:
c         UX  = 1, UY  = 2, UZ  = 3, ROTX= 4, ROTY= 5, ROTZ= 6, AX  = 7, AY  = 8
```

```
c         AZ  = 9, VX  =10, VY  =11, VZ  =12  ****** 13-18 are spares **********
c         ***************  PRES=19, TEMP=20, VOLT=21, MAG =22, ENKE=23, ENDS=24
c         EMF =25, CURR=26  ********* 27-32 are spares ************************
c                                    (curdof(i),i=1,numdof)
c
c  ---      i       1      lenbac   Nodal equivalence table. This table equates
c                                   the number used for storage to the actual
c                                   node number
c                                    (baclst(i),i=1,lenbac)
c
c  TRI   At this point in the file, the triangularized matrix information is
c        stored.  The info is written row by row, and there are two different
c        storage options for writing a row.  If the row being written does not
c        have a constraint equation associated with it, then two records are
c        written to describe the row.  If the row being written has a constraint
c        equation associated with it, then five records are written to describe
c        the row.  Both formats are shown below.  These groupings of two or five
c        records per row will be written a total of nontp times (to include all
c        rows)
c
c  The next two descriptions show the format for a row that does not have a
c  constraint equation associated with it:
c
c  ---   dp/cmp     1     varies    A row of the triangularized matrix.
c
c                                   If keyuns=0, this record will contain the
c                                    non-diagonal terms of this column, the
c                                    diagonal term itself, the normalized F
c                                    term, followed by the reciprocal of the row
c                                    pivot.
c
c                                   If keyuns=1, this record will contain the
c                                    non-diagonal terms of this column, the
c                                    diagonal term itself, the normalized F
c                                    term, the reciprocal of the row pivot,
c                                    followed by the non-diagonal terms of this
c                                    row.
c
c                                   The length of this record will vary (actual
c                                   length is returned from routine BINRD8). If
c                                   kan=3, this record contains complex
c                                   information, otherwise it contains double
c                                   precision information.
c                                    (ktri(i),i=1,n),vload,diag (symmetric)
c                                    (ktri(i),i=1,n),vload,diag,(utri(i),i=1,n)
c                                     (unsymmetric)
c
c  ---      i       1     varies    Triangular matrix row indices. The first
c                                   item signifies what term in the row belongs
c                                   to the pivot. The second term signifies what
c                                   DOF is being eliminated, and the remaining
c                                   items signify the new DOFs being introduced
c                                   (if any). The length of this record will
c                                   vary (actual length is returned from routine
c                                   BINRD8).
c                                    (lll(i),i=1,m)
c
c  The next five descriptions show the format for a row that has a constraint
c  equation associated with it.
c
c  ---      dp      1        2      A flag record, indicating that constraint
c                                   equations are being stored, and the storage
c                                   is as shown here.  Both values are TINY.
c
c  ---      dp      1     varies    Coefficients of the constraint equation.
c                                   The length of this record will vary (actual
c                                   length is returned from routine BINRD8).
c                                    (coeff(i),i=1,n+2)
c
c  ---   dp/cmp     1     varies    A row of the triangularized matrix.
c
c                                   If keyuns=0, this record will contain the
```

```
c                                    non-diagonal terms of this column, the
c                                    diagonal term itself, the normalized F
c                                    term, followed by the reciprocal of the row
c                                    pivot.
c
c                                   If keyuns=1, this record will contain the
c                                    non-diagonal terms of this column, the
c                                    diagonal term itself, the normalized F
c                                    term, the reciprocal of the row pivot,
c                                    followed by the non-diagonal terms of this
c                                    row.
c
c                                   The length of this record will vary (actual
c                                   length is returned from routine BINRD8). If
c                                   kan=3, this record contains complex
c                                   information, otherwise it contains double
c                                   precision information.
c                                    (ktri(i),i=1,n),vload,diag (symmetric)
c                                    (ktri(i),i=1,n),vload,diag,(utri(i),i=1,n)
c                                     (unsymmetric)
c
c  ---      i       1     varies    Triangular matrix row indices. The first
c                                   item signifies what term in the row belongs
c                                   to the pivot. The second term signifies what
c                                   DOF is being eliminated, and the remaining
c                                   items signify the new DOFs being introduced
c                                   (if any). The length of this record will
c                                   vary (actual length is returned from routine
c                                   BINRD8).
c                                    (lll(i),i=1,m)
c
c  ---      i       1        2      This record indicates the end of the 5
c                                   record storage for this row.  It is included
c                                   for situations when the file is being read
c                                   from the bottom up.  Its contents:
c
c                                   -cenum,    n
c
c                                   cenum  - the constraint equation number for
c                                            the constraint equation stored
c                                            above
c                                   n      - the length of a row of the matrix
c                                            being written
c
c  MST      i       1      nmast    This record is present only if nmast > 0.
c   or                              This index is calculated as
c  MS1,                             (N-1)*NUMDOF+DOF, where N is the position
c  MS2                              number of the node in the nodal equivalence
c                                   table, and DOF is the DOF reference number
c                                   given above
```

1.9. Description of the Full Stiffness-Mass File

This section explains the contents of the full file (**jobname.full**).

1.9.1. Standard ANSYS File Header

See Section 1.1.2: The Standard Header for ANSYS Binary Files for a description of this set. File number (Item 1) is 4.

1.9.2. FULL File Format

```
*comdeck,fdfull
c *** ansys(r) copyright(c) 2001
c *** ansys, inc.

c     **********  description of full stiffness-mass file  **********
```

```
c *** mpg fdfull.inc < stff10 slvstr: full file description

      character*8  FULLNM
      parameter  (FULLNM='full    ')

      LONGINT         fullfpL, fullfp
      integer         fullbk,  fullut, wrLdstep, wrSbstep, wrEqiter
      common /fdfull/ fullfpL, fullbk, fullut,
     x                wrLdstep,wrSbstep,wrEqiter
      equivalence  (fullfp,fullfpL)

c   ********** common variable descriptions **********
co  fullfpL      file position on file full
co  fullbk       block number for file full
co  fullut       file unit for file full

c     **********  file format (except for extopt=3,4) **********

c   See fddesc for documentation of how binary files are stored.

c     **********  file format  **********

c        recid tells the identifier for this record.  Not all records will have
c             identifiers -- they are only indicated for those records whose
c             record pointers are stored in the second file header.

c        type tells what kind of information is stored in this record:
c             i - integer
c             dp - double precision
c             cmp - complex

c        nrec tells how many records of this description are found here

c        lrec tells how long the records are (how many items are stored)

c recid    type    nrec    lrec     contents

c  ---      i       1      100      standard ANSYS file header (see binhed for
c                                   details of header contents)

c  ---      i       1       40      .FULL FILE HEADER
c
c                                    fun04,   neqn,  nmrow, nmatrx,    kan,
c                                    wfmax, lenbac, numdof,jcgtrmL,jcgtrmH,
c                                    lumpm, jcgeqn, jcgtrm, keyuns, extopt,
c                                    keyse, sclstf, nxrows,ptrIDXl,ptrIDXh,
c                                   ncefull,ncetrm,ptrENDl,ptrENDh,      0,
c                                        0,      0,      0,      0,      0,
c                                        0,      0,      0,      0,      0,
c                                        0,      0,      0,      0,      0

c NOTE: If fun04 > 0, then the file was created with frontal assembly
c       If fun04 < 0, then the file was created with symbolic assembly;see below
c                     for its format

c -------------------------- frontal assembled file --------------------------

c                                  each item in header is described below:

c                                    fun04  - unit number (full file is 4)
c                                    neqn   - number of equations on file
c                                    nmrow  - number of rows in matrices
c                                    nmatrx - number of matrices on file
c                                    kan    - analysis type
c                                    wfmax  - maximum wavefront
c                                    lenbac - number of nodes
c                                    numdof - number of dofs per node
c                                    jcgtrmL,jcgtrmH - number of coefficients
c                                    lumpm  - lumped mass key
c                                             = 0 - default matrix type
c                                             = 1 - lumped
```

```
c                                   jcgeqn - number of jcg equations
c                                   jcgtrm - pre-8.1 this is the number of
c                                            coefficients in sparse jcg
c                                            storage (otherwise this value
c                                            must be 0 and jcgtrmL,jcgtermH
c                                            must be used)
c                                   keyuns - unsymmetric key
c                                            = 0 - no unsymmetric matrices on
c                                                  file
c                                            = 1 - there is at least one
c                                                  unsymmetric matrix on file
c                                   extopt - mode extraction method
c                                            = 0 - reduced
c                                            = 1 - lumped
c                                            = 3 - unsymmetric Lanczos
c                                            = 4 - damped Lanczos
c                                            = 6 - block Lanczos
c                                   keyse  - superelement key; set if at least
c                                            one superelement
c                                   sclstf - scale factor for matrices
c                                   nxrows - the maximum rank for this solution
c                                   ptrIDXl- pointer to the matrix row indices.
c                                   ptrIDXh- high part of row index pointer
c                                   ncefull- Number of constraint equations on
c                                            the full file
c                                   ncetrm - Total number of terms in the
c                                            constraint equations
c                                   ptrENDl- Low part of 64 bit end of file ptr
c                                   ptrENDh- High part of 64 bit end of file ptr
c                                   0      - position not used

c  ---      i       1      numdof   Degrees of freedom per node
c                                   DOF reference numbers are:
c        UX  = 1, UY  = 2, UZ  = 3, ROTX= 4, ROTY= 5, ROTZ= 6, AX  = 7, AY  = 8
c        AZ  = 9, VX  =10, VY  =11, VZ  =12  ****** 13-18 are spares **********
c        ****************  PRES=19, TEMP=20, VOLT=21, MAG =22, ENKE=23, ENDS=24
c        EMF =25, CURR=26  ********* 27-32 are spares ************************
c                                    (curdof(i),i=1,numdof)

c  ---      i       1      lenbac   Nodal equivalence table. This table equates
c                                   the number used for storage to the actual
c                                   node number
c                                    (baclst(i),i=1,lenbac)

c  NOTE: The next five records are repeated as a group neqn times.
c   When the matrices get written, one row of each matrix is written to the file
c   at a time. i.e. the first row of each matrix is written, then the second row
c   of each matrix, etc. this pattern continues until all the rows of each
c   matrix have been written to the file.  If kan=3, the matrix rows will be
c   complex valued, otherwise they will be double precision values.


c  IDX      i       1      varies   Matrix row indices. The first
c                                   item signifies what term in the row belongs
c                                   to the pivot. The second term signifies what
c                                   DOF is being eliminated, and the remaining
c                                   items signify the new DOFs being introduced
c                                   (if any). The length of this record will
c                                   vary (actual length is returned from routine
c                                   BINRD8)
c                                    (lll(i),i=1,m)

c  ---      i       1      varies   A second level of indexing for the
c                                   matrix. Indicates positions and
c                                   values of terms to be reduced. The length of
c                                   this record will vary (actual length is
c                                   returned from routine BINRD8)
c                                    (index(i),i=1,n) for compressed rows

c  ---   dp/cmp     1      varies   Stiffness matrix.

c                                   If keyuns=0, this record will contain the
```

```
c                                    non-diagonal terms of this column, the
c                                    diagonal term itself, followed by the
c                                    normalized F term.

c                                   If keyuns=1, this record will contain the
c                                    non-diagonal terms of this column, the
c                                    diagonal term itself, the non-diagonal terms
c                                    of this row, followed by the normalized F
c                                    term.

c                                   If lumpm = 1, then the mass for this node is
c                                   located after the F term.  The length of
c                                   this record will vary (actual length is
c                                   returned from routine BINRD8)
c                                    (krow(i),i=1,n),vload,(mass) (symmetric)
c                                    (n-1 column) diag (n-1 row) load (dmass)
c                                     (unsymmetric)

c  ---   dp/cmp     1      varies   Mass matrix.  This record exists only if
c                                   nmatrix > 1.

c                                   If keyuns=0, this record will contain the
c                                    non-diagonal terms of this column, the
c                                    diagonal term itself, followed by the
c                                    normalized F term.

c                                   If keyuns=1, this record will contain the
c                                    non-diagonal terms of this column, the
c                                    diagonal term itself, followed by the
c                                    non-diagonal terms of this row.

c                                   The length of this record will vary (actual
c                                   length is returned from routine BINRD8)
c                                    (mrow(i),i=1,n) (symmetric)
c                                    (n-1 column) diag (n-1 row) (unsymmetric)
c
c                                   If lumpms=1, this record contains one double
c                                   array with diag values

c  ---   dp/cmp     1      varies   Damping matrix.  This record exists only if
c                                   nmatrx > 2.

c                                   If keyuns=0, this record will contain the
c                                    non-diagonal terms of this column, the
c                                    diagonal term itself, followed by the
c                                    normalized F term.

c                                   If keyuns=1, this record will contain the
c                                    non-diagonal terms of this column, the
c                                    diagonal term itself, followed by the
c                                    non-diagonal terms of this row.

c                                   The length of this record will vary (actual
c                                   length is returned from routine BINRD8)
c                                    (ceqn(i),i=1,n) (symmetric)
c                                    (n-1 column) diag (n-1 row) (unsymmetric)

c ------------------------- symbolic assembled file -------------------------

c  ---      i       1       40      .FULL FILE HEADER
c
c                                      fun04,   neqn,   nmrow,  nmatrx,     kan,
c                                      wfmax, lenbac,  numdof, ntermKl, ntermKh,
c                                      lumpm,  nmrow,  ntermK, keyuns,   extopt,
c                                      keyse, sclstf,  nxrows, ptrSTFl, ptrSTFh,
c                                     ncefull,ntermMh, ptrENDl, ptrENDh,ptrIRHSl,
c                                    ptrIRHSh,ptrMASl, ptrMASh, ptrDMPl, ptrDMPh,
c                                      ptrCEl, ptrCEh,  nNodes, ntermMl, ntermDl,
c                                     ptrDOFl,ptrDOFh, ptrRHSl, ptrRHSh, ntermDh

c                                   each item in header is described below:
```

```
c                                   fun04  - negative of the unit number (-4)
c                                   neqn   - number of equations on file
c                                   nmrow  - number of active DOF (neqn-BC)
c                                   nmatrx - number of matrices on file
c                                   kan    - analysis type
c                                   wfmax  - maximum row size
c                                   lenbac - number of nodes
c                                   numdof - number of dofs per node
c                                   ntermKl,ntermKh - number of terms in Stiffness
c                                                     matrix
c                                   lumpm  - lumped mass key
c                                            = 0 - default matrix type
c                                            = 1 - lumped
c                                   ntermK - pre-8.1 this is the number of terms
c                                            in Stiffness matrix (otherwise this
c                                            value must be 0 and ntermKl,ntermKh
c                                            must be used)
c                                   keyuns - unsymmetric key
c                                            = 0 - no unsymmetric matrices on
c                                                  file
c                                            = 1 - there is at least one
c                                                  unsymmetric matrix on file
c                                   extopt - mode extraction method
c                                            = 0 - reduced
c                                            = 1 - lumped
c                                            = 3 - unsymmetric Lanczos
c                                            = 4 - damped Lanczos
c                                            = 6 - block Lanczos
c                                   keyse  - superelement key; set if at least
c                                            one superelement
c                                   sclstf - scale factor for matrices
c                                   nxrows - the maximum rank for this solution
c                                   ptrSTFl,h- pointer to Stiffness matrix
c                                   ncefull- number of CE+CP equations
c                                   ptrENDl- low part of 64 bit end of file ptr
c                                   ptrENDh- high part of 64 bit end of file ptr
c                                   ptrIRHSl,h -pointer to imaginary RHS (F)
c                                   ptrMASl,h - pointer to Mass matrix
c                                   ptrDMPl,h - pointer to Damping matrix
c                                   ptrCEl,h  - pointer to Gt and g matrices
c                                   nNodes    - number of internal Nodes
c                                               considered by symbolic assembly
c                                   ntermMl,h - number of terms in Mass matrix
c                                   ntermDl,h - number of terms in Damping matrix
c                                   ptrDOFl,h - pointer to DOF info
c                                   ptrRHSl,h - pointer to RHS (F)
c                                   0      - position not used

c  ---      i       1      numdof   Degrees of freedom per node
c                                   DOF reference numbers are:
c         UX  = 1, UY  = 2, UZ  = 3, ROTX= 4, ROTY= 5, ROTZ= 6, AX  = 7, AY  = 8
c         AZ  = 9, VX  =10, VY  =11, VZ  =12  ****** 13-18 are spares **********
c         ***************  PRES=19, TEMP=20, VOLT=21, MAG =22, ENKE=23, ENDS=24
c         EMF =25, CURR=26  ********* 27-32 are spares ************************

c  ---      i       1      lenbac   Nodal equivalence table. This table equates
c                                   the number used for storage to the actual
c                                   node number

c Stiffness Matrix. The next two records are repeated as a group neqn times.

c  STF      i       1      varies   Matrix row indices. The last item
c                                   corresponds to the diagonal. The
c                                   length of this record will vary (actual
c                                   length is returned from routine BINRD8)

c  ---   dp/cmp     1      varies   Matrix terms

c                                   If keyuns=0, this record will contain the
c                                    terms before the diagonal.

c                                   If keyuns=1, this record will contain the
```

```
c                                    the entire row.
c
c Load Vector

c  RHS   dp/cmp     1        neqn   Load vector terms.

c Imaginary part of Load Vector

c  IRHS    dp       1        neqn   Imaginary load vector terms.

c DOF information

c  DOF      i       1      nNodes   Nodal extent vector. Number of DOFs at
c                                    each node

c  ---      i       1        neqn   DOF vector. If negative, this DOF
c                                    constrained

c  ---      i       1        neqn   DOFs with imposed values

c  ---   dp/cmp     1      varies   Imposed values

c Mass Matrix.
c    if lumpm = 0:
c      The next two records are repeated as a group neqn times.

c  MAS      i       1      varies   Matrix row indices. The last item
c                                    corresponds to the diagonal. The
c                                    length of this record will vary (actual
c                                    length is returned from routine BINRD8)

c  ---     dp       1      varies   Matrix terms

c    if lumpm = 1:
c  ---     dp       1        neqn   Matrix diagonals

c Damping Matrix. The next two records are repeated as a group neqn times.

c  DMP      i       1      varies   Matrix row indices. The last item
c                                    corresponds to the diagonal. The
c                                    length of this record will vary (actual
c                                    length is returned from routine BINRD8)

c  ---     dp       1      varies   Matrix terms

c G matrix if ncefull > 0.

c  CE       i       1     ncefull   List of slave DOFs

c  ---     dp       1     ncefull   g vector (constant terms)

c  ---      i       1           4   Header; 1=nRows, 2=nRows, 3=1, 4=0

c  ---      i       1       nRows   Vector of 1's

c  ---      i       1       nRows   Number of non-zero terms in each row

c   Repeat for each row:

c  ---      i       1      varies   Column indices

c  ---     dp       1      varies   Column values

c  Meaning of K11, K12, and G matrices:
c     Given
c         [K]{x} = {F}
c     subject to the constraints
c           {x1} = [G]{x2} + {g}
c     where {x1} are the slave DOFs, {x2} the master DOFs

c     This results in
c         [K*]{x2} = {F*}
```

```
c     where
c         [K*] = [G]'[K11][G] + [G]'[K12] + [K21][G] + [K22]
c         {F*} = [G]'{f1} + {f2} - [G]'[K11]{g} - [K21]{g}

c complex version of {F*} decomposed into, we assume G' is always real
c and g could be complex denoted as g' == (g,gx) :
c         G' K11' g' = G' (K11,M11)*(g,gx)
c                    = G' [K11*g - M11*gx, M11*g + K11*gx]

c         K21' *g'   = (K21,M21)*(g,gx)
c                    = (K21*g- M21*gx, K21*gx + M21*g)
```

Chapter 2: Accessing Binary Data Files

2.1. Accessing ANSYS Binary Files

The following section explains the routines you need to read, write, or modify an ANSYS binary file. This collection of routines (called BINLIB) resides on your ANSYS distribution media. The BINLIB library is in the dynamic link library **\Program Files\Ansys Inc\V100\ANSYS\custom\misc\<*platform*>\binlib.dll** (on Windows systems (where <*platform*> is a directory that uniquely identifies the hardware platform version)) or the shared library **/ansys_inc/v100/ansys/customize/misc/<*platform*>/libbin.so** on UNIX systems (**libbin.sl** on HP systems).

Your distribution media also includes sample FORTRAN source files which employ the BINLIB library:

bintrd.F	The bintrd subroutine reads and prints the contents of an ANSYS binary file
bintwr.F	The bintwr subroutine copies an ANSYS binary file to a new file
bintst.F	The bintst program calls the bintwr and bintrd subroutines as an example of how to use the binlib library to print the contents of a file, copy the original file to a new file, and then print the contents of the new file. Routine bintst has no inputs or outputs. It requires use of the bintcm common. (For more information, see the descriptions of the bintrd and bintwr routines later in this chapter.)

These files reside in the subdirectory **\Program Files\Ansys Inc\V100\ANSYS\custom\misc\<*platform*>** (on Windows systems) or **/ansys_inc/v100/ansys/customize/misc** (on UNIX systems). To compile and link the BINTST program, a **makefile** procedure has been provided in the **\Program Files\Ansys Inc\V100\ANSYS\custom\misc\<*platform*>** subdirectory on windows or run the **bintst.link** procedure in the **/ansys_inc/v100/ansys/customize/misc** subdirectory on UNIX.

2.1.1. Access Routines to Results and Substructure Files

Demonstration programs for reading and writing ANSYS results and substructure files are included on the installation media:

- `ResRdDemo`
- `ResWrDemo`
- `rdsubs`
- `wrtsub`
- `rdfull`

On Windows systems:

The FORTRAN source for these programs is located in **\Program Files\Ansys Inc\V100\ANSYS\custom\misc** and the files are named **ResRdDemo.F**, **ResWrDemo.F**, **rdsubs.F**, **wrtsub.F**, and **rdfull.F**.

To link these demonstration programs, use the **\Program Files\Ansys Inc\V100\ANSYS\custom\misc\<*platform*>\rdrwrt.bat** procedure file and specify the program that you want to build on the command line. Valid command line options are `ResRdDemo`, `ResWrDemo`, `rdsubs`, `wrtsub`, **rdfull**, and `userprog`. For example, to build the program to read a results file, type:

```
\Program Files\Ansys Inc\V100\ANSYS\custom\misc\<platform>\rdrwrt  ResRdDemo
```

Appropriate files will then be copied from **\Program Files\Ansys Inc\V100\ANSYS\custom\misc\<*platform*>** to your working directory, compiled, and linked. The resulting executable will also be placed in your current working directory.

Use the userprog command line option when writing your own customized program, naming the routine **userprog.F**. The resulting executable will be named **userprog.exe**. When `userprog` is used, no files are copied from **\Program Files\Ansys Inc\V100\ANSYS\custom\misc\<*platform*>** to your working directory.

These files will be loaded onto your system only if you performed a custom installation and chose to install the customization tools.

On UNIX systems:

The FORTRAN source for these programs is located in **/ansys_inc/v100/ansys/custom/misc** and the files are named **ResRdDemo.F**, **ResWrDemo.F**, **rdsubs.F**, **wrtsub.F**, and **rdfull.F**.

To link these demonstration programs, use the **/ansys_inc/v100/ansys/customize/misc/rdrwrt.link** procedure file and specify the program that you want to build on the command line. Valid command line options are `ResRdDemo`, `ResWrDemo`, `rdsubs`, `wrtsub`, **rdfull**, and `userprog`. For example, to build the program to read a results file, type:

```
/ansys_inc/v100/ansys/customize/misc/rdrwrt.link  ResRdDemo
```

Appropriate files will then be copied from **/ansys_inc/v100/ansys/customize/misc** to your working directory, compiled, and linked. The resulting executable will also be placed in your current working directory. Procedure files are available in the **/ansys_inc/v100/ansys/bin** directory to run these programs, once linked. The procedure files are named `ResRdDemo100`, `ResWrDemo100`, `rdsubs100`, `wrtsub100`, and **rdfull100**.

Use the userprog command line option when writing your own customized program, naming the routine **userprog.F**. The resulting executable will be named **userprog.e100**. When `userprog` is used, no files are copied from **/ansys_inc/v100/ansys/customize/misc** to your working directory. The procedure file is named `userprog100`.

These files will be loaded onto your system only if you performed a custom installation and chose to install the customization tools.

2.1.2. Characteristics of ANSYS Binary Files

Before accessing ANSYS binary files, you need to know certain file characteristics:

1. An ANSYS binary file is a direct access, unformatted file. You read or write a record by specifying (as a number) what location to read or write.
2. Before the ANSYS program actually writes data to a file on a disk, it uses buffers to store data in memory until those buffers become full. A block number designates these buffers. Most access routines use this block number.
3. By default, ANSYS files are external files. The standardized "external" format the files use enables you to transport them across different computer systems.
4. In addition to file names, ANSYS uses file numbers to identify the files. File handles and other information are associated with the file numbers.
5. Some binary files contain data values that point to the start of certain data (for example, the start of the data steps index table record). Both the ANSYS program and external binary files access routines use these pointers to locate data on the various binary files.

6. All data is written out as 32-bit integers. Double-precision data and pointers, therefore, take up two integer words. To create a 64-bit pointer from the two 32-bit integers, use the function largeIntGet.

2.1.3. Viewing Binary File Contents

To view the contents of certain ANSYS binary files, you issue the command **/AUX2** or choose menu path **Utility Menu>File>List>Binary Files** or **Utility Menu>List>File>Binary Files**. (You can do so only at the Begin level.) The ANSYS program then enters its binary file dumping processor, AUX2, and dumps the binary file record by record.

AUX2 does not use the data pointers discussed in item 5 above. It uses record numbers to locate the binary file data to dump. Although the information that AUX2 provides includes the pointer, using the pointer alone will not get you that information. To get it, you must correlate the pointer and the record number by trial and error.

2.1.4. Abbreviations

The input and output for the routines discussed in this chapter are described with the following abbreviations:

- *Type* of variable is one of the following:

  int - integer
  dp - double-precision
  log - logical (true or false)
  char - character

- *Size* of variable is one of the following:

  sc - scalar variable
  ar(*n*) - array of size *n*

- *Intent* of variable is one of the following:

  in - input only
  out - output only
  inout - both an input and an output variable

2.1.5. binini (Initializing Buffered Binary I/O Systems)

```
*deck,binini
      subroutine binini (iott)
c *** primary function: initialize buffered binary i/o system
c --- This routine is intended to be used in standalone programs.
c --- This routine should not be linked into the ANSYS program.

c *** Notice - This file contains ANSYS Confidential information ***

c  input arguments:
c     iott      (int,sc,in)       - output unit number for error output

c  output arguments:   none
```

2.1.6. Function sysiqr (Retrieving the Status of a File)

```
*deck,sysiqr
      function sysiqr (nunit,fname,lname_in,inqr_in)

c *** primary function: do a file system inquire (system dependent)
```

```
c *** Notice - This file contains ANSYS Confidential information ***

c  input arguments:
c    variable (typ,siz,intent)    description
c    nunit    (int,sc,in)       - fortran unit number (used only for inqr='O')
c    fname    (chr,sc,in)       - name of file
c    lname_in (int,sc,in)       - length of file name (characters, max=50)
c    inqr_in  (chr,sc,in)       - character key for information requested
c                                  = 'E' - return whether file exists
c                                      sysiqr = 1 - file exists
c                                             = 0 - file does not exist
c                                             < 0 - error occured
c                                  = 'O' - return whether file is open
c                                      sysiqr = 1 - file is open
c                                             = 0 - file is closed
c                                             < 0 - error occured
c                                  = 'N' - return unit number of file
c                                      sysiqr > 0 - unit number for file
c                                             = 0 - file not assigned to a unit
c                                             < 0 - error occured

c  output arguments:
c    sysiqr   (int,func,out)    - the returned value of sysiqr is based on
c                                   setting of inqr
```

2.1.7. Function biniqr8 (Retrieving System-Dependent Parameters)

```
*deck,biniqr8
      function biniqr8 (nblk,key)
c *** primary function: get data about a block i/o buffer
c --- This routine is intended to be used in standalone programs.
c --- This routine should not be linked into the ANSYS program.

c *** Notice - This file contains ANSYS Confidential information ***

c  input arguments:
c     nblk      (int,sc,in)       - the block number for the inquiry
c                                   or zero (see below)
c     key       (int,sc,in)       - key for information requested
c                       nblk = 0 - return information about system/file
c                                key = 1 - return system block size
c                                    = 2 - return number of integers per dp
c                                    = 3 - return filename length
c                                      5 = return integers per LONG
c                       nblk > 0 - return information about this block
c                                key = 1 - return fortran unit number
c                                    = 2 - return number of blocks in file
c                                    = 3 - return length of page (32 bit words)
c                                    = 4 - return open status
c                                          0 - file close
c                                          1 - file open
c                                    = 5 - return file format
c                                          0 - internal format
c                                          1 - external format
c                                    = 6 - return read/write status
c                                          0 - both read & write
c                                          1 - read
c                                          2 - write
c                                    = 7 - return current position on file
c                                    = 8 - return maximum length of file
c                                            (in words)
c                                    = 9 - return starting word for this page
c                                            in buffer

c  output arguments:
c     biniqr    (int,func,out)    - the returned value of biniqr is based on
c                                     setting of nblk and key
```

2.1.8. Function binset (Opening a Blocked Binary File or Initializing Paging Space)

```
*deck,binset
      function binset (nblk,nunit,ikeyrw,istart,paglen,npage,
     x                 pname,nchar,kext,Buffer4)
c *** primary function: initialize paging space for a blocked binary file.
c                       binset should be used to open a blocked file
c                       before binrd8 or binwrt8 are used.  binclo should
c                       be used to close the file.
c --- This routine is intended to be used in standalone programs.
c --- This routine should not be linked into the ANSYS program.

c *** Notice - This file contains ANSYS Confidential information ***

c  input arguments:
c     nblk      (int,sc,in)       - block number (1 to BIO_MAXFILES max)
c     nunit      (int,sc,in)       - fortran unit number for the file
c                                    (if 0, bit bucket)
c     ikeyrw    (int,sc,in)       - read/write flag
c                                    =  0 - both read & write
c                                    =  1 - read
c                                    =  2 - write
c                                    =  9 - read only
c     istart    (int,sc,in)       - starting location in buffer array
c                                    usually 1 for nblk=1, paglen*npage+1
c                                    for nblk=2,etc.
c     paglen    (int,sc,in)       - page length in integer*4 words for external
c                                    files
c                                    paglen should always be a multiple of
c                                    512 words for efficiency
c     npage     (int,sc,in)       - number of pages (1 to BIO_MAXBLOCKS max)
c     pname     (chr,ar(*),in)    - name of the file
c     nchar     (int,sc,in)       - number of characters in the file name(not
c                                     used)
c     kext      (int,sc,in)       - no longer used, always external format
c     Buffer4   (i4, ar(*),inout) - work array for paging, should be
c                                    dimensioned to paglen*npage*nblk (max)

c  output arguments:
c     binset    (int,func,out)    - error status
c                                    = 0 - no error
c                                    <>0 - error occurred
c     Buffer4   (i4, ar(*),inout) - work array for paging
```

2.1.9. Subroutine bintfo (Defining Data for a Standard ANSYS File Header)

```
*deck,bintfo
      subroutine bintfo (title,jobnam,units,code)
c *** primary function:    set information necessary for binhed
c --- This routine is intended to be used in standalone programs.
c --- This routine should not be linked into the ANSYS program.

c
c *** Notice - This file contains ANSYS Confidential information ***
c
c     typ=int,dp,log,chr,dcp   siz=sc,ar(n)   intent=in,out,inout
c
c  input arguments:
c     variable (typ,siz,intent)    description
c     title    (chr*80,ar(2),in) - main title and 1st subtitle
c     jobnam   (chr*8,sc,in)     - jobname
c     units    (int,sc,in)       - units
c                                    = 0 - user defined units
c                                    = 1 - SI
c                                    = 2 - CGS
c                                    = 3 - British, using feet
c                                    = 4 - British, using inches
```

```
c     code     (int,sc,in)       - code defining 3rd party vendor
c                                  (contact ANSYS, Inc. for code assignment)
c
c  output arguments:
c     none
c
```

2.1.10. Subroutine binhed (Writing the Standard ANSYS File Header)

```
*deck,binhed
      subroutine binhed (nblk,nunit,filpos,buffer)
c *** primary function:    put standard header on a binary file, all
c                          permanent binary files should have this header
c *** secondary functions: return the first data position
c --- This routine is intended to be used in standalone programs.
c --- This routine should not be linked into the ANSYS program.

c *** Notice - This file contains ANSYS Confidential information ***

c  input arguments:
c     nblk     (int,sc,in)       - block number of open binary file
c                                  (as defined with subroutine binset)
c     nunit    (int,sc,in)       - the unit number for this file
c     buffer   (int,ar(*),inout) - work array for paging, should be the
c                                  same array as used in binset

c  output arguments:
c     filpos   (int,sc,out)      - the position after the header
c     buffer   (int,ar(*),inout) - work array for paging

c   **********  ANSYS standard header data description (100 words)  **********
c  loc   no. words   contents
c   1       1        fortran unit number
c   2       2        file format
c                    = 0 - internal format
c                    = 1 - external format
c   3       1        time in compact form (ie 130619 is 13:06:19)
c   4       1        date in compact form (ie 19981023 is 10/23/1998)
c   5       1        units
c                    = 0 - user defined units
c                    = 1 - SI
c                    = 2 - CSG
c                    = 3 - British, using feet
c                    = 4 - British, using inches
c   6       1        User_Linked
c  10       1        revision in text format ' 5.0' (inexc4)
c  11       1        date of revision release for this version
c  12       3        machine identifier - 3 4-character strings
c  15       2        jobname - 2 4-character strings
c  17       2        product name - 2 4-character strings
c  19       1        special version label - 1 4-character string
c  20       3        user name - 3 4-character strings
c  23       3        machine identifier - 3 4-character strings
c  26       1        system record size at file write
c  27       1        maximum file length
c  28       1        maximum record number
c  31       8        jobname - 8 4-character strings
c  41      20        main title - 20 4-character strings
c  61      20        first subtitle - 20 4-character strings
c  95       1        split point of file
c 97-98     2        LONGINT of file size at write
```

2.1.11. Subroutine binrd8 (Reading Data from a Buffered File)

```
*deck,binrd8
      subroutine binrd8 (nblk,LongLocL,leng,ivect,kbfint,Buffer4)

c **********  buffer read routine  **********
```

```
c *** Notice - This file contains ANSYS Confidential information ***

c  input arguments:
c   nblk   (int,sc,in)     - block number.  see fd___(i.e. fdtri for tri
c
c                            (as defined with subroutine bioset)
c   LongLocL(LONG,sc,inout)- location in integer*4 words of the startin
c                            position on the file.
c   leng   (int,sc,inout)  - number of words to read into ivect. (must be
c                            less or equal to dimension given to ivect in
c                            the calling routine).  if ivect is to be used
c                            as integers, use as is.  if ivect is to be
c                            used for double precision numbers, it must be
c                            increased by multiplying it by INTPDP.
c                            if negative, skip record and do not return
c                            data(results).
c                            data(results).
c   Buffer4 (i4, ar(*),inout) - work array for paging, should be the
c                                  same array as used in binset

c  output arguments:
c   LongLocL(LONG,sc,inout)- location in integer*4 words of the current
c                            position on the file.  It is updated after
c                            each read operation
c   leng   (int,sc,inout)  - tells you how many items it actually read(in
c                            integer words).
c                            if zero, end of file(error case)
c   ivect  (int,ar(*),out) - results (can be either integer or double
c                            precision in the calling routine)
c   kbfint (int,sc,out)    - key for type(used only for AUX2 dump)
c                           = 0  double precision data
c                           > 0  integer data(usually the same as leng)
c   Buffer4 (i4,ar(*),inout) - work array for paging
```

Versions of **binrd8**/**binwrt8** exist without the "8" suffix (**binrd**/**binwrt**) that take a regular integer for the second argument. These subroutines, therefore, cannot address large files where the file position exceeds 2**31. Use the **binrd8**/**binwrt8** versions for any new programs.

2.1.12. Subroutine binwrt8 (Writing Data to a Buffered File)

```
*deck,binwrt8
      subroutine binwrt8 (nblk,LongLocL,leng,ivect,kbfint,Buffer4)
c *** primary function: buffer write routine
c --- This routine is intended to be used in standalone programs.
c --- This routine should not be linked into the ANSYS program.

c *** Notice - This file contains ANSYS Confidential information ***

c  input arguments:
c   nblk   (int,sc,in)    - block number.  see fd___(i.e. fdtri for tri
c
c   LongLocL(LONG,sc,inout)- location in integer words of the starting
c                            position on the file.
c   leng   (int,sc,in)    - number of words to read from ivect. (must be
c                            less or equal to dimension given to ivect in
c                            the calling routine).  if ivect is to be used
c                            as integers, use as is.  if ivect is to be
c                            used for double precision numbers, it must be
c                            increased by multiplying it by INTPDP.
c   ivect  (int,ar(*),in) - data to be written onto the file(can be either
c                            integer or double precision in the calling
c                            routine)
c   kbfint (int,sc,in)    - key for type(used only for AUX2 dump)
c                           = 0  double precision data
c                           > 0  integer data(usually the same as leng)
c   Buffer4 (int,ar(*),inout) - work array for paging, should be the
c                                  same array as used in binset on this
c                                  block
```

```
c  output arguments:
c   LongLocL(LONG,sc,inout)- location in integer words of the current
c                            position on the file.  It is updated after
c                            each write operation
c   ivect  (int,ar(*),out)- vector containing record to be written
c   Buffer4 (int,ar(*),inout) - work array for paging
```

Versions of **binrd8**/**binwrt8** exist without the "8" suffix (**binrd**/**binwrt**) that take a regular integer for the second argument. These subroutines, therefore, cannot address large files where the file position exceeds 2**31. Use the **binrd8**/**binwrt8** versions for any new programs.

2.1.13. Subroutine exinc4 (Decoding an Integer String into a Character String)

```
*deck,exinc4
      subroutine exinc4 (ichext,chin,n)
c primary function: decode externally formatted integer versions of 4-character
c                   strings to plain 4-character strings (used to convert data
c                   from externally formatted files to data for interally
c                   formatted files)
c
c *** Notice - This file contains ANSYS Confidential information ***
c
c  input arguments:
c     ichext    (int,ar(n),in)    - externally formatted integer form of
c                                   4-character strings
c     n         (int,sc,in)       - number of strings to convert
c
c  output arguments:
c     chin     (char,ar(n),out)  - strings in character form
c
```

2.1.14. Subroutine inexc4 (Coding a Character String into an Integer String)

```
*deck,inexc4
      subroutine inexc4 (chin,ichext,n)
c primary function: encode plain 4-character strings into externally formatted
c                   integer versions of 4-character strings (used to convert
c                   data from internally formatted files to data for
c                   externally formatted files)
c
c *** Notice - This file contains ANSYS Confidential information ***
c
c  input arguments:
c     chin      (char,ar(n),out)  - strings in character form
c     n         (int,sc,in)       - number of strings to convert
c
c  output arguments:
c     ichext    (int,ar(n),in)    - externally formatted integer form of
c                                   4-character strings
c
```

2.1.15. Subroutine binclo (Closing or Deleting a Blocked Binary File)

```
*deck,binclo
      subroutine binclo (nblk,pstat,Buffer4)
c *** primary function: close a blocked file, every block/file opened with
c                      binset should be closed with binclo
c *** secondary function: the file can be deleted by specifying 'D' in pstat
c --- This routine is intended to be used in standalone programs.
c --- This routine should not be linked into the ANSYS program.

c *** Notice - This file contains ANSYS Confidential information ***

c  input arguments:
c     nblk      (int,sc,in)       - the block number to close
```

```
c                                  (as defined with subroutine binset)
c     pstat     (chr,sc,in)       - keep or delete flag
c                                  = 'K' - keep file
c                                  = 'D' - delete file
c     Buffer4   (int,ar(*),inout) - work array for paging, should be the
c                                  same array as used in binset

c  output arguments:
c     Buffer4    (int,ar(*),inout) - work array for paging
```

2.1.16. Subroutine largeIntGet (Converting Two Integers into a Pointer)

```
*deck,largeIntGet
      function largeIntGet (small,large)

c primary function:    Convert two short integers into a long integer

c object/library:  res

c *** Notice - This file contains ANSYS Confidential information ***

c  input arguments:
c     small    (int,sc,in)       - least significant part
c     large    (int,sc,in)       - most significant part

c  output arguments:
c     largeIntGet  (LONGINT,sc,out)  - 64 bit integer
```

2.2. Demonstration Routines

The demonstration routines demonstrate several ways to use the binary file access routines provided with ANSYS. The programs described below (all available on your distribution media; see Section 2.1: Accessing ANSYS Binary Files for their location) demonstrate other tasks that the binary access routines can do.

2.2.1. Program bintst (Demonstrates Dumping a Binary File and Copying It for Comparison Purposes)

The `bintst` program dumps a binary file with the name **file.rst** to the screen. It then takes that file, copies it to a new file, **file2.rst**, and dumps the new file to the screen for comparison purposes.

2.2.1.1. Common Variables:

Variable	Type, Size, Intent	Description
iout	int, sc, comm	The output unit number
intpdp	int, sc, comm	The number of integers per double precision word
lenfrm	int, sc, comm	The number of characters in the filename
reclng	int, sc, comm	The system record length

Note — The bintst program is not part of the **binlib.a** library. It is included here only to aid you.

2.2.2. Subroutine bintrd (Demonstrates Printing a Dump of File Contents)

```
*deck,bintrd
      subroutine bintrd (pname)
c *** primary function: bin file dump utility
c
c *** Notice - This file contains ANSYS Confidential information ***
c
```

```
c *** ansys(r) copyright(c) 2000
c *** ansys, inc.
c
c     typ=int,dp,log,chr,dcp   siz=sc,ar(n)   intent=in,out,inout
c
c  input arguments:
c     variable  (typ,siz,intent)    description
c     pname     (chr,sc,in)       - name of binary file which is to
c                                   be dumped to the screen
c
c  output arguments:
c     none.
c
c  common variables:
c     iout      (int,sc,comm)     - output unit number
c     intpdp    (int,sc,comm)     - number of integers per double precision word
c     lenfnm    (int,sc,comm)     - number of characters in the filename
c     reclng    (int,sc,comm)     - system record length
c
c                           NOTE: bintrd is not part of binlib.a.  it is
c                                 included only as an aid to users.
c
```

Note — The `bintrd` routine and the `bintwr` routine described below are not part of **binlib.a**. This chapter includes it only to aid you. You can find the source for this routine on the ANSYS distribution media.

Both subroutines require the following common:

```
COMMON/BINTCM/ IOUT,INTPDP,LENFNM,RECLNG
```

- *`Iout`* is the output unit number.
- *`Intpdp`* is the number of integers per double precision word.
- *`Lenfnm`* is the number of characters in the filename.
- *`Reclng`* is the system record length.

2.2.3. Subroutine bintwr (Demonstrates Copying Binary File Contents)

```
*deck,bintwr
      subroutine bintwr (pname,nname)
c *** primary function: bin file copy utility
c
c *** Notice - This file contains ANSYS Confidential information ***
c
c *** ansys(r) copyright(c) 2000
c *** ansys, inc.
c
c     typ=int,dp,log,chr,dcp   siz=sc,ar(n)   intent=in,out,inout
c
c  input arguments:
c     variable  (typ,siz,intent)    description
c     pname     (chr,sc,in)       - name of binary file which is to be copied
c
c  output arguments:
c     variable  (typ,siz,intent)    description
c     nname     (chr,sc,out)      - name of new binary file which is a copy
c                                   of pname
c  common variables:
c     iout      (int,sc,comm)     - output unit number
c     intpdp    (int,sc,comm)     - number of integers per double precision word
c     lenfnm    (int,sc,comm)     - number of characters in the filename
c     reclng    (int,sc,comm)     - system record length
c
c                           NOTE: bintwr is not part of binlib.a.  it is
c                                 included only as an aid to users.
c
```

2.2.4. Program wrtsub (Demonstrates Writing an ANSYS Substructure File)

```
*deck,wrtsub
      program wrtsub

c primary function:    demonstrates use of binary access routines
c secondary function:  write an ANSYS substructure file

c *** Notice - This file contains ANSYS Confidential information ***
c
c *** ansys(r) copyright(c) 2000
c *** ansys, inc.
c
c  ********************************************************************
c  *  Program to write ANSYS substructure file with usrsub.  To be     *
c  *  used as a base for 3rd party companies to create their routines  *
c  *  for writing the file.                                            *
```

2.2.5. Program rdsubs (Demonstrates Reading a Substructure File)

Subroutine `rdsubs` demonstrates how you read an ANSYS substructure file. This demonstration program can handle up to *MAXNODE* nodes and *MAXDOF* degrees of freedom.

```
*deck,rdsubs
      program rdsubs
c primary function:    demonstrates use of binary access routines
c secondary function:  read an ANSYS substructure file

c *** Notice - This file contains ANSYS Confidential information ***
c
c *** ansys(r) copyright(c) 2000
c *** ansys, inc.
c
c  ********************************************************************
c  *  Reads a substructure file.  To be used as base for 3rd party     *
c  *  development of routines for reading ANSYS substructure files.    *
c  *  This demonstration program can handle up to:                     *
c  *   MAXNODE Nodes and MAXDOF DOFs                                   *
c  ********************************************************************
```

2.2.6. Program rdfull (Demonstrates Reading and Reformatting the .FULL File)

Program `rdfull` demonstrates how to read and reformat the **.FULL** file. ANSYS writes the full file if the **PSOLVE**,ELFORM, **PSOLVE**,ELPREP, **PSOLVE**,TRIANG sequence is used. You can also use the **WRFULL** command.

If you want to use the free stiffness and mass matrices, make sure that there are no constraints on your model.

```
*deck,rdfull
      program  rdfull
c *** primary function:    demonstrates use of binary access routines
c *** secondary function:  Read and reformat full file
c
c *** ansys(r) copyright(c) 2000
c *** ansys, inc.
c
c  NOTICE - A new assembly process, termed 'symbolic assembly', has
c           replaced the old assembly process, termed 'frontal
c           assembly', and is now the default assembly process for
c           most analyses.  This program demonstrates how to read
c           and reformat the .FULL file that was created using
c           frontal assembly or symbolic assembly.  ANSYS writes the
c           .FULL file if the PSOLVE,ELFORM
c                             PSOLVE,ELPREP
c                             PSOLVE,TRIANG
```

```
c           sequence is used.  ANSYS will also write the .FULL file
c           when the sparse, ICCG, or JCG solver is used, as well as
c           when most mode extraction methods are used.

c    Be sure to set up for modal  ANTYPE,2
c    and full subspace            MODOPT,SUBSP,nmode,0,0, ,OFF
c       (nmode is not used - it can be any value

c      If the free-free stiffness and mass matrices are desired,
c        make sure there are no constraints on the model.
```

2.2.7. Program ResRdDemo (Demonstrates Reading a Results File)

Program `ResRdDemo` demonstrates how to read a results file using the results file access routines. The file must be named **test.rst** and the file contents are written to the screen.

This file resides in the subdirectory **\Program Files\Ansys Inc\V100\ANSYS\custom\misc\<*platform*>** (on Windows systems) or **/ansys_inc/v100/ansys/customize/misc** (on UNIX systems).

2.2.8. Program ResWrDemo (Demonstrates Writing a Results File)

Program `ResWrDemo` demonstrates how to write an ANSYS-readable results file. This file resides in the subdirectory **\Program Files\Ansys Inc\V100\ANSYS\custom\misc\<*platform*>** (on Windows systems) or **/ansys_inc/v100/ansys/customize/misc** (on UNIX systems).

2.3. Results File Access Routines

You can use the low-level routines in described in Section 2.1: Accessing ANSYS Binary Files to retrieve data from the results file. Alternatively, you can use the routines described in this section that retrieve the data specific to the format of the results file. These routines are included on the installation CD (compressed); you can install them on your system by performing a custom installation (as described in the *ANSYS Installation and Configuration Guide* for your platform).

These files reside in the subdirectory **\Program Files\Ansys Inc\V100\ANSYS\custom\misc\<*platform*>** (on Windows systems) or **/ansys_inc/v100/ansys/customize/misc** (on UNIX systems). See Section 2.1.1: Access Routines to Results and Substructure Files for information on compiling and linking these routines.

2.3.1. Overview of the Routines

For each data record in the results file, routines exist that:

- Read the record index and allocate space for the data. These are named `ResRd`*`record`*`Begin`, where *`record`* is a descriptive name of the record, e.g., `ResRdNodeBegin`
- Read the data itself. These are named `ResRd`*`record`*, e.g., `ResRdNode`
- Deallocate space for the data. These are named `ResRd`*`record`*`End`, e.g., `ResRdNodeEnd`

Below is a complete listing of all the routines with the indentation indicating the required nested calling sequence:

```
function ResRdBegin (Nunit,Lunit,Fname,ncFname,Title,JobName,
  subroutine ResRdGeomBegin (MaxType,MaxReal,MaxCsys)
    subroutine ResRdTypeBegin (NumType)
      function ResRdType (itype,ielc)
    subroutine ResRdTypeEnd
    subroutine ResRdRealBegin (NumReal,NumPerReal)
      function ResRdReal (iReal,Rcon)
    subroutine ResRdRealEnd
    subroutine ResRdCsysBegin (NumCsys)
      function ResRdCsys (iCsys,Csys)
```

```
    subroutine ResRdCsysEnd
    subroutine ResRdNodeBegin
      function ResRdNode (iNode,xyzang)
    subroutine ResRdNodeEnd
    subroutine ResRdElemBegin
      function ResRdElem (iElem,nodes,ElemData)
    subroutine ResRdElemEnd
  subroutine ResRdGeomEnd
  function ResRdSolBegin (key,lstep,substep,ncumit,kcmplx,time,
    subroutine ResRdDispBegin
      function ResRdDisp (node,Disp)
    subroutine ResRdDispEnd
    subroutine ResRdRforBegin (nRForce)
      function ResRdRfor (node,idof,value)
    subroutine ResRdRforEnd
    subroutine ResRdBCBegin (BCHeader)
      subroutine ResRdFixBegin (BCHeader,nFixed)
        function ResRdFix (node,idof,value)
      subroutine ResRdFixEnd
      subroutine ResRdForcBegin (BCHeader,nForces)
        function ResRdForc (node,idof,value)
      subroutine ResRdForcEnd
    subroutine ResRdBCEnd
    subroutine ResRdEresBegin
      function ResRdEstrBegin  (iElem)
        function ResRdEstr (iStr,Str)
      subroutine ResRdEstrEnd
    subroutine ResRdEresEnd
  subroutine ResRdSolEnd
subroutine ResRdEnd
```

These routines are contained in the file **ResRd.F**. See the demonstration routine **ResRdDemo.F** on the distribution medium for an example of the usage of these routines.

The memory allocation scheme is described in Memory Management Routines in the *Guide to ANSYS User Programmable Features*.

The following sections describe the data-reading routines. See the file **ResRd.F** and its corresponding include deck **ResRd.inc** for listings of the corresponding Begin/End routines.

2.3.2. ResRdBegin (Opening the File and Retrieving Global Information)

```
*deck,ResRdBegin
      function ResRdBegin (Nunit, Lunit, Fname, ncFname, Title, JobName,
     x                     Units, NumDOF, DOF, UserCode,
     x                     MaxNode, NumNode, MaxElem, NumElem,
     x                     MaxResultSet,NumResultSet)
c primary function:    Open result file and return global information

c object/library: ResRd

c  input arguments:
c     Nunit    (int,sc,in)       - Fortran Unit number for file (ANSYS uses 12)
c     Lunit    (int,sc,in)       - Current print output unit (usually 6 <STDOUT>)
c     Fname    (ch*(ncFname),sc,in) - The name (with extension) for the file
c     ncFname  (int,sc,in)       - Number of characters in Fname

c  output arguments:
c     Title    (ch*80,ar(2),out) - Title and First subtitle
c     JobName  (ch*8,sc,out)     - Jobname from file
c     Units    (int,sc,out)      - 0, unknown  1, SI  2, CSG
c                                  3, U.S. Customary - foot
c                                  4, U.S. Customary - inch
c                                  6, MPA
c     NumDOF   (int,sc,out)      - Number of DOF per node
c     DOF      (int,ar(*),out)   - The DOFs per node
c     UserCode (int,sc,out)      - Code for this application
c     MaxNode  (int,sc,out)      - Maximum node number used
```

```
c     NumNode  (int,sc,out)      - Number of nodes used
c     MaxElem  (int,sc,out)      - Maximum element number used
c     NumElem  (int,sc,out)      - Number of elements used
c     MaxResultSet (int,sc,out)  - Maximum number of result sets (usually 1000)
c     NumResultSet (int,sc,out)  - Number of result sets on file
c     ResRdBegin (int,sc,out)    - 0, successful  other, error in file open
```

2.3.3. ResRdGeomBegin (Retrieving Global Geometry Information)

```
*deck,ResRdGeomBegin
      subroutine ResRdGeomBegin (MaxType, MaxReal, MaxCsys)
c primary function:    Read Geometry Header Record

c object/library: ResRd

c  input arguments:  none

c  output arguments:
c     MaxType  (int,sc,out)       - Maximum element type
c     MaxReal  (int,sc,out)       - Maximum real constant set number
c     MaxCsys  (int,sc,out)       - Maximum coordinate system number
```

2.3.4. ResRdType (Retrieving Element Types)

```
*deck,ResRdType
      function ResRdType (itype,ielc)
c primary function:    Read an element type record

c object/library: ResRd

c  input arguments:
c     itype    (int,sc,on)         - Element type number

c  output arguments:  none
c     ielc     (int,ar(IELCSZ),out)- Element characteristics
c     ResRdType (int,sc,out        - number of words read
```

2.3.5. ResRdReal (Retrieving Real Constants)

```
*deck,ResRdReal
      function ResRdReal (iReal,Rcon)
c primary function:    Read real constant record

c object/library: ResRd

c  input arguments:
c     iReal    (int,sc,in)             - Real set number

c  output arguments:  none
c     Rcon      (dp,ar(ResRdReal),out) - Real Constants
c     ResRdReal (int,sc,out)           - Number of real constants in set
```

2.3.6. ResRdCsys (Retrieving Coordinate Systems)

```
*deck,ResRdCsys
      function ResRdCsys (iCsys,Csys)
c primary function:    Read a coordinate system record

c object/library: ResRd

c  input arguments:
c     iCsys     (int,sc,in)           - Coordinate system number

c  output arguments:
c     Csys      (dp,ar(ResRdCsys),out)- Coordinate system description
```

```
c     ResRdCsys (int,sc,out)          - Number of values

c  output arguments:
```

2.3.7. ResRdNode (Retrieving Nodal Coordinates)

```
*deck,ResRdNode
      function ResRdNode (iNode,xyzang)
c primary function:    Get a node

c object/library: ResRd

c  input arguments:
c     iNode     (int,sc,in)     - Node number

c  output arguments:
c     xyzang    (dp,ar(6),out)  - x,y,z,thxy,thyz,thzx for node
c     ResRdNode (int,sc,out)    - >0, node exists
```

2.3.8. ResRdElem (Retrieving Elements)

```
*deck,ResRdElem
      function ResRdElem (iElem, nodes, ElemData)
c primary function:    Read an element

c object/library: ResRd

c  input arguments:
c     iElem    (int,sc,in)       - The element number

c  output arguments:
c     ResRdElem(int,sc,out)      - Number of nodes
c     nodes    (int,ar(n),out)   - Element nodes
c     ElemData (int,ar(10),out)  - Element information
c                                    mat    - material reference number
c                                    type   - element type number
c                                    real   - real constant reference number
c                                    elnum  - element number
c                                    esys   - element coordinate system
c                                    death  - death flag
c                                             = 0 - alive
c                                             = 1 - dead
c                                    solidm - solid model reference
c                                    shape  - coded shape key
c                                    pexcl  - P-Method exclude key
c                                    0      - Not used
```

2.3.9. ResRdSolBegin (Retrieving Result Set Location)

```
*deck,ResRdSolBegin
      function ResRdSolBegin (key,lstep,substep,ncumit,kcmplx,time,
     x                        Title,DofLab)
c primary function:    Read the solution header records

c object/library: ResRd

c  input arguments:
c     key      (int,sc,in)       - 1, find by lstep/substep
c                                  2, find by ncumit
c                                  3, find by time
c     lstep    (int,sc,in/out)   - Load step number
c     substep  (int,sc,in/out)   - Substep of this load step
c     ncumit   (int,sc,in/out)   - Cumulative iteration number
c     kcmplx   (int,sc,in)       - 0, Real solution   1, Imaginary solution
c     time     (int,sc,in/out)   - Current solution time

c  output arguments:
```

```
c     Title    (ch*80,ar(5),out)  - Title and 4 subtitles
c     DofLab   (ch*4,ar(nDOF),out)- Labels for DOFs
c     ResRdSolBegin (int,sc,out)  - 0, requested solution set found
c                                   1, not found
```

2.3.10. ResRdDisp (Retrieving Nodal Solution)

```
*deck,ResRdDisp
      function ResRdDisp (node,Disp)
c primary function:    Retrieve a nodal displacement

c object/library: ResRd

c  input arguments:
c     node     (int,sc,in)       - Node number

c  output arguments:  none
c     Disp     (dp,ar(nDOF),out) - Displacements
c     ResRdDisp(int,sc,out)      - Number of displacements
```

2.3.11. ResRdRfor (Retrieving Reaction Solution)

```
*deck,ResRdRfor
      function ResRdRfor (node,idof,value)
c primary function:    Retrieve a reaction force

c object/library: ResRd

c  input arguments:
c     node     (int,sc,in)       - External node number
c     idof     (int,sc,in)       - Internal dof number

c  output arguments:
c     value     (dp,sc,in)       - Value of reaction force
c     ResRdRfor (int,sc,out)     - Number of returned values (0 or 1)
```

2.3.12. ResRdFix (Retrieving Applied Nodal Constraints)

```
*deck,ResRdFix
      function ResRdFix (node,idof,value)
c primary function:    Retrieve a constraint value

c object/library: ResRd

c  input arguments:
c     node     (int,sc,in)       - External node number
c     idof     (int,sc,in)       - Internal dof number

c  output arguments:
c     value    (dp,ar(4),in)     - Real,Imag, RealOld,ImagOld
c     ResRdFix (int,sc,out)      - Number of returned values (0 or 4)
```

2.3.13. ResRdForc (Retrieving Applied Nodal Loads Solution)

```
*deck,ResRdForc
      function ResRdForc (node,idof,value)
c primary function:    Retrieve an applied force value

c object/library: ResRd

c  input arguments:
c     node     (int,sc,in)       - External node number
c     idof     (int,sc,in)       - Internal dof number

c  output arguments:
```

```
c     value    (dp,ar(4),in)     - Real,Imag, RealOld,ImagOld
c     ResRdForc (int,sc,out      - Number of returned values (0 or 4)
```

2.3.14. ResRdEstr (Retrieving Element Solutions)

```
*deck,ResRdEstr
      function ResRdEstr (iStr,Str)
c primary function:   Get an element's results

c object/library: ResRd


c  input arguments:
c     iStr    (int,sc,in)       - element record number (1-25)

c  output arguments:
c     ResRdEstr (int,sc,out)       - Number of element values
c     Str       (dp,ar(nStr),out)  - element values
```

Chapter 3: Using CDREAD and CDWRITE

3.1. Using the CDREAD Command

The **CDREAD** command and its GUI equivalent, **Main Menu> Preprocessor> Archive Model> Read**, read a file of model and database information into the ANSYS database. The commands and menu paths listed below define the input data that the ANSYS program requires to solve a model. If the file you are reading into the database via **CDREAD** or **Main Menu> Preprocessor> Archive Model> Read** does not contain all of the required input data, you can use these commands or menu paths to define that data in the preprocessor. For detailed information about the commands, see the *ANSYS Commands Reference*.

In the following list, commands or menu paths shown with an asterisk (*) are the minimum requirements for a solution.

Command	Equivalent Menu Path	Definition
/PREP7*	**Main Menu> Preprocessor***	Enters the general preprocessor
ET*	**Main Menu> Preprocessor> Element Type> Add/Edit/Delete***	Defines the element types
MP*	**Main Menu> Preprocessor> Material Props***	Defines the material properties
MPTEMP	**Main Menu> Preprocessor> Material Props> Material Models**	Defines a table of material temperatures
MPDATA	**Main Menu> Preprocessor> Material Props> Material Models**	Defines a table of material properties
TB	**Main Menu> Preprocessor> Material Props> Material Models**	Defines nonlinear material data, some element data, or both
R	**Main Menu> Preprocessor> Real Constants> Add/Edit/Delete**	Defines element real constants
LOCAL	**Utility Menu> WorkPlane> Local Coordinate Systems> Create Local CS> At Specified Loc**	Defines a local coordinate system
N*	**Main Menu> Preprocessor> Modeling> Create> Nodes> In Active CS** or **Main Menu> Preprocessor> Modeling> Create> Nodes> On Working Plane**	Defines a nodal location
E or **EN***	**Main Menu> Preprocessor> Modeling> Create> Elements> Auto Numbered> Thru Nodes** or **Main Menu> Preprocessor> Modeling> Create> Elements> User Numbered> Thru Nodes**	Defines an element. (You must set the correct TYPE, MAT, REAL, and ESYS pointers before defining the element.)
ANTYPE	**Main Menu> Preprocessor> Loads> Analysis Type> New Analysis**	Defines the analysis type. See the *ANSYS Commands Reference* for details.
M	**Main Menu> Preprocessor> Loads> Master DOFs> User Selected> Define**	Defines master degrees of freedom
ACEL	**Main Menu> Preprocessor> Loads> Define Loads> Apply> Structural> Inertia> Gravity**	Defines the acceleration vector
D*	**Main Menu> Preprocessor> Loads> Define Loads> Apply>** ***constraint***	Defines degree of freedom constraints
F	**Main Menu> Preprocessor> Loads> Define Loads> Apply>** ***force type*****> On** ***entity***	Defines nodal forces

Command	Equivalent Menu Path	Definition
SF or **SFE**	**Main Menu> Preprocessor> Loads> Define Loads> Apply>** ***surface load type*** **> On** ***entity***	Defines surface loads on element faces
BF or **BFE**	**Main Menu> Preprocessor> Loads> Define Loads> Apply>** ***load type*** **> On** ***entity***	Defines body forces on elements

3.1.1. Tips for Reading Files with CDREAD

The following list describes practices to follow when reading files with **CDREAD** or its menu path equivalent:

- If you place the command **/NOPR** at the beginning of the file and **/GOPR** at the end of the file, you can suppress printing of the file and speed up the time to read it.
- Use the **/COM** command to add comments to the file. If the file contains the **/NOPR** command, **/COM** provides status information about what is being read in. For example:

```
/COM, READING NODES
N,1,.....
 ,
 ,

/COM, READING ELEMENTS
EN,1,.....
```

- Group elements with the same attributes (TYPE, REAL, MAT, ESYS). *Do not* use a separate **MAT**, **REAL**, **TYPE**, and **ESYS** command for each element.
- Use **Utility Menu> File> Change Title** or the **/TITLE** command to add a title to graphics displays and printouts.
- Use **Utility Menu> File> Read Input From** or the **/INPUT** command at the Begin level to input the file.

3.2. Using the CDWRITE Command

To export a model from the ANSYS program to another application, use menu path **Main Menu> Preprocessor> Archive Model> Write** or the **CDWRITE** command within the general preprocessor, PREP7. This produces a coded database file called **Jobname.cdb**. You specify the jobname using **Utility Menu> File> Change Jobname** or the **/FILNAM** command. If you supply no jobname, the ANSYS program uses the default name "file".

The **Jobname.cdb** file contains selected geometry (nodes and elements), load items, and other model data in terms of ANSYS input commands. (For a complete list of data in the file, see the **CDWRITE** description in the *ANSYS Commands Reference*.) You can convert this information to a format compatible with the program into which you are importing it. The next few pages describe special considerations and commands you may need to do this conversion.

Note — Files created by the **CDWRITE** command have the active coordinate system set to Cartesian (**CSYS**, 0).

ANSYS may create parameters in the **CDWRITE** file that start with an underscore (_), usually an "_z." Such parameters are for ANSYS internal use and pass information to the ANSYS GUI.

3.2.1. Customizing Degree of Freedom Labels: the /DFLAB Command

The ANSYS program uses a set of default labels for the degrees of freedom. You use these labels when entering boundary conditions, or ANSYS uses the labels when writing the **Jobname.cdb** file.

You can change the labels to reflect the degrees of freedom of the other program by issuing the command **/DFLAB**. If you are customizing the DOF labels, **/DFLAB** must be the first command you enter within the ANSYS program. You may want to include the command in your **START.ANS**file. You can use **/DFLAB** only at the Begin processing level.

/DFLAB assigns or reassigns the "displacement" and "force" labels in the ANSYS DOF list. For example, degree of number 1 is predefined to have a displacement label of UX and a force label of FX, but you can assign new labels to this DOF using by issuing **/DFLAB**. Changing predefined labels generates a warning message.

The format for the **/DFLAB** command is:

```
/DFLAB,NDOF,LabD,LabF
```

NDOF
ANSYS degree of freedom number (1 to 32)

LabD
Displacement degree of freedom label to be assigned (up to four characters)

LabF
Force label to be assigned (up to four characters)

You can also use **/DFLAB** to assign labels to spare degree of freedom numbers. Spare displacement and force labels are from 13 to 18 and from 27 to 32. All other DOF numbers are predefined, as follows:

DOF Number	Corresponding Displacement Label	Corresponding Force Label
1	UX	FX
2	UY	FY
3	UZ	FZ
4	ROTX	MX
5	ROTY	MY
6	ROTZ	MZ
7	AX	CSGX
8	AY	CSGY
9	AZ	CSGZ
10	VX	VFX
11	VY	VFY
12	VZ	VFZ
19	PRES	FLOW
20	TEMP	HEAT
21	VOLT	AMPS
22	MAG	FLUX
23	ENKE	NPKE
24	ENDS	NPDS
25	EMF	CURT
26	CURR	VLTG

3.3. Coded Database File Commands

In the coded database file **Jobname.CDB**, most ANSYS commands have the same format they have elsewhere. (See the *ANSYS Commands Reference* for command-specific information.) However, the format for some commands differs slightly in the **Jobname.CDB** file. The format for these commands is described below.

The **CDWRITE** command has an UNBLOCKED and a BLOCKED option. The UNBLOCKED option will write all data out in command format. the default BLOCKED option will write certain data items in a fixed format, especially those which could potentially contain large amounts of data, such as nodal data.

3.3.1. CE Command

The **CE** command defines the constant term in a constraint equation. The command format in **Jobname.CDB** is:

```
CE,R5.0,Type,LENGTH,NCE,CONST
```

Type
The type of data to be defined. DEFI is the valid label.

LENGTH
The total number of variable terms in the constraint equation.

NCE
The constraint equation reference number.

CONST
The constant term of the equation.

Another version of the **CE** command defines the variable terms in a constraint equation. You must issue this version of the command after the **CE** command described above. This command repeats until all terms are defined.

The alternate format for the **CE** command is:

```
CE,R5.0,Type,N1,Dlab1,C1,N2,Dlab2,C2
```

Type
The type of data to be defined. NODE is the valid label.

N1
The node number of the next term.

Dlab1
The DOF label of *N1*.

C1
The coefficient of *N1*.

N2
The node number of the next term.

Dlab2
The DOF label of *N2*.

C2
The coefficient of *N2*.

3.3.2. CP Command

The **CP** command defines a coupled node set. You repeat the command until all nodes are defined. The command format in **Jobname.CDB** is:

```
CP,R5.0,LENGTH,NCP,Dlab,N1,N2,N3,N4,N5,N6,N7
```

LENGTH
The total number of nodes in the coupled set

NCP
The coupled node reference number

Dlab
The degree of freedom label for the set

N1,*N2*,*N3*,*N4*,*N5*,*N6*,*N7*
The next seven node numbers in the coupled set

3.3.3. CMBLOCK Command

The **CMBLOCK** command defines the entities contained in a node or element component. The command format in **Jobname.CDB** is:

```
CMBLOCK,Cname,Entity,NUMITEMS
Format
```

Cname
Eight character component name.

Entity
Label identifying the type of component (NODE or ELEMENT).

NUMITEMS
Number of items written.

Format
Data descriptors defining the format. For **CMBLOCK** this is always (8i10).

The items contained in this component are written at 10 items per line. Additional lines are repeated as needed until all *NumItems* are defined. If one of the items is less than zero, then the entities from the item previous to this one (inclusive) are part of the component.

3.3.4. EBLOCK Command

The **EBLOCK** command defines a block of elements. The command syntax is:

```
EBLOCK, NUM_NODES, Solkey
Format
```

NUM_NODES
The number of nodes to be read in the first line of an element definition.

Solkey
The solid model key. The element is part of a solid model if the keyword SOLID appears here. When *Solkey* = SOLID, Field 8 (the element shape flag) may be left at zero, and Field 9 is the number of nodes defining this element.

Format
Data descriptors defining the format. This must be 19i7.

The format of the element "block" is as follows:

- Field 1 - The material number.
- Field 2 - The element type number.
- Field 3 - The real constant number.
- Field 4 - The section ID attribute (beam section) number.
- Field 5 - The element coordinate system number.
- Field 6 - The birth/death flag.
- Field 7 - The solid model reference number.
- Field 8 - The element shape flag.
- Field 9 - The number of nodes defining this element if *Solkey* = SOLID; otherwise, Field 9 = 0.
- Field 10 - The exclude key (p-elements).
- Field 11 - The element number.
- Fields 12-18 - The node numbers. The next line will have the additional node numbers if there are more than eight.

The final line of the block will be a -1 in field 1.

If you are in the GUI, the **EBLOCK** command must be contained in an externally prepared file and read into ANSYS (i.e., **CDREAD**, **/INPUT**, etc.).

3.3.5. EDCADAPT Command

The **EDCADAPT** command specifies adaptive meshing control for explicit dynamics analysis. The command format in **Jobname.CDB** is:

```
EDCADAPT,R5.3,FREQ,TOL,OPT,MAXLVL,BTIME,DTIME,LCID,ADPSIZE,ADPASS,IREFLG,ADPENE,ADPTH,MAXEL
```

FREQ
The time interval between adaptive mesh refinement.

TOL
The adaptive angle tolerance (in degrees).

OPT
The adaptivity option.

MAXLVL
The maximum number of mesh refinement levels.

BTIME
The birth time to begin adaptive meshing.

DTIME
The death time to end adaptive meshing.

LCID
The curve ID defined by **EDCURVE**

ADPSIZE
The minimum element size to be adapted, based on the element edge length.

ADPASS
The one-pass or two-pass adaptivity option.

IREFLG
The uniform refinement level flag.

ADPENE
Adaptive mesh flag for starting adaptivity when approaching (positive *ADPENE*) or penetrating (negative *ADPENE*) the tooling surface

ADPTH
Absolute shell thickness level below which adaptivity should begin.

MAXEL
The maximum number of elements at which adaptivity will be terminated.

NOTE: This command is also listed in the *ANSYS Commands Reference*. The format listed here contians information specific to the CDREAD/CDWRITE file.

3.3.6. EDCGEN Command

The **EDCGEN** command is used to define a contact definition for explicit dynamics. The command format in **Jobname.CDB** is:

```
EDCGEN,R5.3,Option,Cont,Targ,Lkey,FS,FD,DC,VC,VDC,V1,V2,V3,V4,BTIME,DTIME,BOXID1,BOXID2
```

Option
The label identifying the contact behavior.

Cont
The contact surface, identified by component name, part ID, or part assembly ID.

Targ
The target surface, identified by component name, part ID, or part assembly ID.

Lkey
A key identifying the meaning of *Cont* and *Targ* (component, part or part assembly).

FS
The static friction coefficient.

FD
The dynamic friction coefficient.

DC
The exponential decay coefficient.

VC
The coefficient of viscous friction.

VDC
The viscous damping coefficient in percent of critical damping.

V1,V2,V3,V4
Additional input for some contact types. See **EDCGEN** in the *ANSYS Commands Reference* for more information.

BTIME
The birth time for which conatct definition will become active.

DTIME
The death time for which conatct definition will become inactive.

BOXID1
Contact volume as defined using EDBOX

BOXID2
Target volume as defined using EDBOX

NOTE: This command is also listed in the *ANSYS Commands Reference*. The format listed here contians information specific to the CDREAD/CDWRITE file.

3.3.7. EDCURVE Command

The **EDCURVE** command is used to define a curve for an explicit dynamics analysis. The command format in **Jobname.CDB** is:

```
EDCURVE,R5.3,Option,LCID,Length,0.0,Par1,Par2
```

Option
The **EDCURVE** command option. The only valid option is "ADD."

LCID
The curve ID.

Length
The number of data values for the abcissa array (*Par1*) and the ordinate array (*Par2*).

Par1
The abcissa values, repeat *Length* number of times.

Par2
The ordinate values, repeat *Length* number of times.

NOTE: This command is also listed in the *ANSYS Commands Reference*. The format listed here contians information specific to the CDREAD/CDWRITE file.

3.3.8. EDDRELAX Command

The **EDDRELAX** command activates initialization to a prescribed geometry or dynamic relaxation for the explicit analysis. The command format in **Jobname.CDB** is:

```
EDDRELAX,R5.4,Option,NRCYCK,IRELAL,EDTTL,DRTOL,DFFCTR,DRTERM,TSSFDR
```

Option
EDDRELAX command option. Valid options are "ANSYS" (relaxation is based on the implicit analysis, see the **EDDRELAX** command in the *ANSYS Commands Reference*) or "DYNA," where the relaxation parameters are controlled within the LS-DYNA analysis. The following arguments are valid for *Option*= DYNA only.

NRCYCK
The number of iterations between the convergence checks.

IRELAL
Automatic control based on Papadrakakis not active (0) or active (1).

EDTTL
The convergence tolerance when automatic control is used.

DRTOL
The convergence tolerance.

DFFCTR
The dynamic relaxation factor.

DRTERM
The termination time for dynamic relaxation.

TSSFDR
The scale factor for each computed time step.

NOTE: This command is also listed in the *ANSYS Commands Reference*. The format listed here contians information specific to the CDREAD/CDWRITE file.

3.3.9. EDLCS Command

The **EDLCS** command is used to define a local coordinate system for explicit dynamics. The command format in **Jobname.CDB** is:

```
EDLCS,R5.3,Option,CID,X1,Y1,Z1,X2,Y2,Z2,X3,Y3,Z3
```

Option
The **EDLCS** command option. The only valid option is "ADD."

CID
The coordinate system ID.

X1,*Y1*,*Z1*,
The X,Y,Z coordinate of a point on the local X-axis.

X2,*Y2*,*Z2*,
The X,Y,Z coordinate of a point on the local X-Y plane.

X3,*Y3*,*Z3*,
The X,Y,Z coordinate of the local origin.

NOTE: This command is also listed in the *ANSYS Commands Reference*. The format listed here contians information specific to the CDREAD/CDWRITE file.

3.3.10. EDLOAD Command

The **EDLOAD** command is used to define loading conditions for explicit dynamics. The command format in **Jobname.CDB** is:

```
EDLOAD,R5.3,Option,Lab,KEY,Cname,Length,PHASE,Par1,Par2,LCID,SCALE,BTIME,DTIME
```

Option
The **EDLOAD** command option. The only valid option is "ADD."

Lab
The load labels.

Key
The coordinate system number defined by EDLCS or the element face number for the pressure loading.

Cname
The name of the existing component or part number to which this load will be applied.

Length
The number of data values for the time array (*Par1*) and the load array (*Par2*).

Phase
Phase of the analysis in which the load curve is to be used.

Par1
The time values, with the number of values in the string defined by the *Length* argument (above).

Par2
The load values, with the number of values in the string defined by the *Length* argument (above).

LCID
The curve ID, created using the **EDCURVE** command. If *LCID* is nonzero, then *Length*= 1, and *Par1* and *Par2* will be equal to 0.

Scale
The Scale Factor applied to the load curve.

Btime
The birth time.

Dtime
The death time.

NOTE: This command is also listed in the *ANSYS Commands Reference*. The format listed here contians information specific to the CDREAD/CDWRITE file.

3.3.11. EDPREAD Command

The **EDPREAD** command is used to internally write the part information to the Jobname.CDB file for explicit dynamics. Prior to Release 8.0, the command format in **Jobname.CDB** is:

```
EDPREAD,R5.4,Nmat,Npart
Type, Mat, Real, Used
```

Nmat
The number of materials.

Npart
Number of parts, and also, the number of times to repeat the second *Type*,*Mat*,*Real*,*Used* input line.

Type
The element type number.

Mat
The material number.

Real
The real constant set number.

Used
The flag indicating if the part is used (1), or not used (0).

For Release 8.0 and beyond, the command format is:

```
EDPREAD,R8.0,Nmat,Npart,Part ID
Type, Mat, Real, Used
```

Nmat
The number of materials.

Npart
Number of parts, and also, the number of times to repeat the second *Type*,*Mat*,*Real*,*Used* input line.

PartID
The part number.

Type
The element type number.

Mat
The material number.

Real
The real constant set number.

Used
The flag indicating how many elements use *PartID*. If USED = 0, *PartID* is not used.

3.3.12. EDWELD Command

The **EDWELD** command is used to define a spotweld or a generalized weld for explicit dynamics.

There are two command formats (for spot and generalized welds). The command format for the spotweld appears in **Jobname.CDB** as follows:

```
EDWELD,R5.3,Option,NWELD,N1,N2,SN,SS,EXPN,EXPS
```

Option
The **EDWELD** command option. The only valid option is "ADD."

NWELD
The spotweld ID number.

N1
The node number of the first node connected by the spotweld.

N2
The node number of the second node connected by the spotweld.

SN
The normal force at the spotweld failure.

SS
The shear force at the spotweld failure.

EXPN
The exponent for spotweld normal force.

EXPS
The exponent for spotweld shear force.

The command format for the generalized weld appears in **Jobname.CDB** as follows:

```
EDWELD,R5.3,Option,NWELD,CNAME,,SN,SS,EXPN,EXPS,EPSF,TFAIL,NSW,CID
```

Option
The **EDWELD** command option. The only valid option is "ADD."

NWELD
The generalized weld ID number.

CNAME
The name of the node component.

SN
The normal force at the weld failure.

SS
The shear force at the weld failure.

EXPN
The exponent for weld normal force.

EXPS
The exponent for weld shear force.

EXPF
The effective plastic strain at ductile failure.

TFAIL
The time of failure of the weld.

NSW
The number of spotwelds for the generalized weld.

CID
The coordinate system ID as defined by the **EDLCS** command.

NOTE: This command is also listed in the *ANSYS Commands Reference*. The format listed here contians information specific to the CDREAD/CDWRITE file.

3.3.13. EN Command

The **EN** command is used to define an element . If an element contains more than eight nodes, the **EN** command is repeated until all nodes are defined. The command format in **Jobname.CDB** is:

```
EN,R5.5,Type,NUMN,I1,I2,I3,I4,I5,I6,I7,I8
```

The type of data to be defined. Valid labels are "ATTR" (read in element attributes), and "NODE" (read in nodes defining the element).

NUMN
The number of nodes.

I1,*I2*,*I3*,*I4I5*,*I6*,*I7*,*I8*
The integer values to be read:

- If *Type* is ATTR, the integer values are the element attributes. Attributes are in the order: NUMN,MAT,TYPE,REAL,SECNUM,ESYS,NUMELEM,SOLID,DEATH,EXCLUDE
- If *Type* is NODE, the integer values are the node numbers.

3.3.14. LOCAL Command

The **LOCAL** command defines a local coordinate system. The command format in **Jobname.CDB** is:

```
LOCAL,R5.0,Type,NCSY,CSYTYP,VAL1,VAL2,VAL3
```

Type
The type of data to be defined. Valid labels are LOC (read in system origin), ANG (read in rotation angles), and PRM (read in system parameters).

NCSY
The coordinate system reference number.

CSYTYP
The coordinate system type (0, 1, 2, or 3).

VAL1,*VAL2*,*VAL3*
Values to be read:

- If *Type* is LOC, values are the system origin in global Cartesian coordinates.
- If *Type* is ANG, values are the rotation angles in degrees.

- If *Type* is PRM, values are the first and second parameters of the system.

3.3.15. M Command

The **M** command defines a master degree of freedom. The command format in **Jobname.CDB** is:

```
M,R5.0,NODE,Dlab
```

NODE
The node number

Dlab
The DOF label

3.3.16. MPDATA Command

The **MPDATA** command defines a material property data table. You repeat the command until all properties are defined. The command format in **Jobname.CDB** is:

```
MPDATA,R5.0,LENGTH,Lab,MAT,STLOC,VAL1,VAL2,VAL3
```

LENGTH
The total number of temperatures in the table.

Lab
The material property label. See the **MP** command description in *ANSYS Commands Reference* for valid labels.

MAT
The material reference number.

STLOC
The starting location in the table for the next three property values.

VAL1,*VAL2*,*VAL3*
Property values assigned to three locations in the table starting at *STLOC*.

3.3.17. MPTEMP Command

The **MPTEMP** command defines a temperature table. You repeat the command until all temperature values are defined. The command format in **Jobname.CDB** is:

```
MPTEMP,R5.0,LENGTH,STLOC,TEMP1,TEMP2,TEMP3
```

LENGTH
The total number of temperatures in the table

STLOC
The starting location in the table for the next three temperature values

TEMP1,*TEMP2*,*TEMP3*
Temperatures assigned to three locations in the table starting at *STLOC*

3.3.18. N Command

If the UNBLOCKED option is used with the **CDWRITE** command, then the **N** command defines a node. This is also the method used for defining nodes in **.CDB** files before ANSYS 5.4. The command format in **Jobname.CDB** is:

```
N,R5.3,Type,NODE,SOLID,PARM,VAL1,VAL2,VAL3
```

Type
The type of data to be defined. Valid labels are LOC (read in coordinates) and ANG (read in rotation angles).

NODE
The node number.

SOLID
The solid model reference key. Not present for *Type*= ANG.

PARM
Line parameter value (0 if not on line). Not present for *Type*= ANG.

VAL1,*VAL2*,*VAL3*
Values to be read:

- If *Type* is LOC, values are the coordinates in the global Cartesian system.
- If *Type* is ANG, values are the rotation angles in degrees.

3.3.19. NBLOCK Command

The **NBLOCK** command defines a block of nodes. This is the recommended method for inputting nodes into the ANSYS data base. The command syntax is:

```
NBLOCK, NUMFIELD, Solkey
Format
```

NUMFIELD
The number of fields in the blocked format.

Solkey
The solid model key. The node is part of a solid model if the keyword SOLID appears here.

Format
Data descriptors defining the format. This must be (3i8,6e16.9).

The format of the node "block" is as follows:

- Field 1 - Node number.
- Field 2 - The solid model entity (if any) in which the node exists.
- Field 3 - The line location (if the node exists on a line).
- Field 4 - 6 - The nodal coordinates.
- Field 7 - 9 - The rotation angles.

Only the last nonzero coordinate/rotation is output; any trailing zero values are left blank.

The final line of the block is always an **N** command using a -1 for the node number.

The following example shows a typical **NBLOCK** formatted set of node information. Note that this example has no rotational data. It contains only the first six fields.

```
NBLOCK,6
(i8,6e16,9)
       1   6.21299982    0.625999987  -1.019883480E-07
       2   6.14472103    0.625999987       0.156284466
       3   6.21271753    0.625999987   1.096193120E-02
.
.
.
  151464    0    0  5.85640764   -0.442010075   1.911501959E-02
```

```
151465    0    0  5.88715029   -0.442010075   7.201258256E-08
151466    0    0  5.85541868   -0.442010075   7.201258256E-08
 N, R5.3,LOC, -1
```

If you are in the GUI, the **NBLOCK** command must be contained in an externally prepared file and read into ANSYS (i.e., **CDREAD**, **/INPUT**, etc.).

3.3.20. R Command

The **R** command defines a real constant set. You repeat the command until all real constants for this set are defined. The command format in **Jobname.CDB** is:

```
R,R5.0,NSET,Type,STLOC,VAL1,VAL2,VAL3
```

NSET
The real constant set reference number.

Type
The type of data to be defined. LOC is the valid label.

STLOC
The starting location in the table for the next three constants.

VAL1,*VAL2*, *VAL3*
Real constant values assigned to three locations in the table starting at *STLOC*.

3.3.21. RLBLOCK Command

The **RLBLOCK** command defines a real constant set. The real constant sets follow each set, starting with Format1 and followed by one or more Format2's, as needed. The command format is:

```
RLBLOCK,NUMSETS,MAXSET,MAXITEMS,NPERLINE
Format1
Format2
```

NUMSETS
The number of real constant sets defined

MAXSET
Maximum real constant set number

MAXITEMS
Maximum number of reals in any one set

NPERLINE
Number of reals defined on a line

Format1
Data descriptor defining the format of the first line. For the **RLBLOCK** command, this is always (2i8,6g16.9). The first i8 is the set number, the second i8 is the number of values in this set, followed by up to 6 real constant values.

Format2
Data descriptors defining the format of the subsequent lines (as needed); this is always (7g16.9).

The real constant sets follow, with each set starting with Format1, and followed by one or more Format2's as needed.

3.3.22. SECBLOCK Command

The **SECBLOCK** command retrieves all mesh data for a user-defined beam section as a block of data. You repeat the command for each beam section that you want to read. The command format is:

```
SECBLOCK
Format1
Format2
Format3
```

Format1
: The First Line section. The first value is the number of nodes, and the second is the number of cells.

Format2
: The Cells Section. The first 9 values are the cell connectivity nodes. The 10th (last) value is the material ID (**MAT**).

Format3
: The Nodes Section. This section contains as many lines as there are nodes. In this example, there are 27 nodes, so a total of 27 lines would appear in this section. Each node line contains the node's boundary flag, the Y coordinate of the node, and the Z coordinate of the node. Currently, all node boundary flags appear as 0s in a cell mesh file. Because all node boundary flags are 0, **SECBLOCK** ignores them when it reads a cell mesh file.

Sample User Section Cell Mesh File

Following is a sample excerpt from a custom section mesh file for a section with 27 nodes, 4 cells, and 9 nodes per cell:

First Line:

```
27   4
```

Cells Section:

```
      1   3   11   9    2   6    10   4    5    2
      7   9   23   21   8   16   22   14   15   1
      9   11  25   23   10  18   24   16   17   1
      11  13  27   25   12  20   26   18   19   1
```

Nodes Section:

```
...
0     0.0          0.0
0     0.025        0.0
0     0.05         0.0
0     5.0175       0.0
0     19.98        10.00
0     20.00        10.00
...
```

3.3.23. SFBEAM Command

The **SFBEAM** command defines a surface load on selected beam elements. Remaining values associated with this specification are on a new input line with a (4f16.9) format. The command format in **Jobname.CDB** is:

```
SFBEAM,ELEM,LKEY,Lab,R5.0,DIOFFST,DJOFFST
```

ELEM
: The element number.

LKEY
: The load key associated with these surface loads.

Lab
: A label indicating the type of surface load. PRES (for pressure) is the only valid label.

DI0FFST
Offset distance from node I.

DJ0FFST
Offset distance from node J.

3.3.24. SFE Command

The **SFE** command defines a surface load. Values associated with this specification are on a new input line with a (4f16.9) format. The command format in **Jobname.CDB** is:

```
SFE,ELEM,LKEY,Lab,KEY,R5.0
```

ELEM
The element number.

LKEY
The load key associated with this surface load.

Lab
A label indicating the type of surface load: Valid labels are:

- PRES (pressure)
- CONV (convection)
- HFLU (heat flux)
- IMPD (impedance)
- SEL (substructure load vector)
- SELV (S. E. load vectors)
- CHRG (charge density)

KEY
A value key. If it is 1, the values are real (film coefficient if convection). If it is 2, values are imaginary (bulk temperature if convection).

Chapter 4: ANSYS Graphics File Format

4.1. Modifying ANSYS Graphics Files

Some ANSYS users may wish to translate ANSYS graphics files to other formats (such as Encapsulated PostScript or AI). If you plan to translate graphics files, this chapter provides some information to help you:

- Source code, with comments, for a program called PLOT33 which plots all of the plots on the coded plot file. You can link this program with user-supplied Calcomp HCBS software (plot, plots, symbol) or other software that uses the Calcomp subroutine protocol. For example, Hewlett-Packard's ISPP product, Tektronix's Preview product, and Veratec's Versaplot product use the Calcomp protocol.

  To work with other plotter software, you may need to remove the calls to the Calcomp subroutine (plot, plots, and symbol) and substitute the subroutine calls for the plotter software being used.

  Note — Because ANSYS customers use a wide variety of plotters, ANSYS, Inc. can not verify the use of the PLOT33 program with all plotter types.

- A description of the format for the ANSYS neutral graphics file. This listing excludes format information for z-buffered graphics, but the PLOT33 program *does* include a section on z-buffered graphics.

4.2. Pixmap Format for Graphic Display Files

The ANSYS graphics display is *KPX* pixels high by *KPX* * 1.33 pixels wide.

KPX is the resolution specified by the **/GFILE**,SIZE command (where SIZE is the pixel resolution) or by choosing menu path **Utility Menu>PlotCtrls>Redirect Plots>To File**. Default resolution is 800.

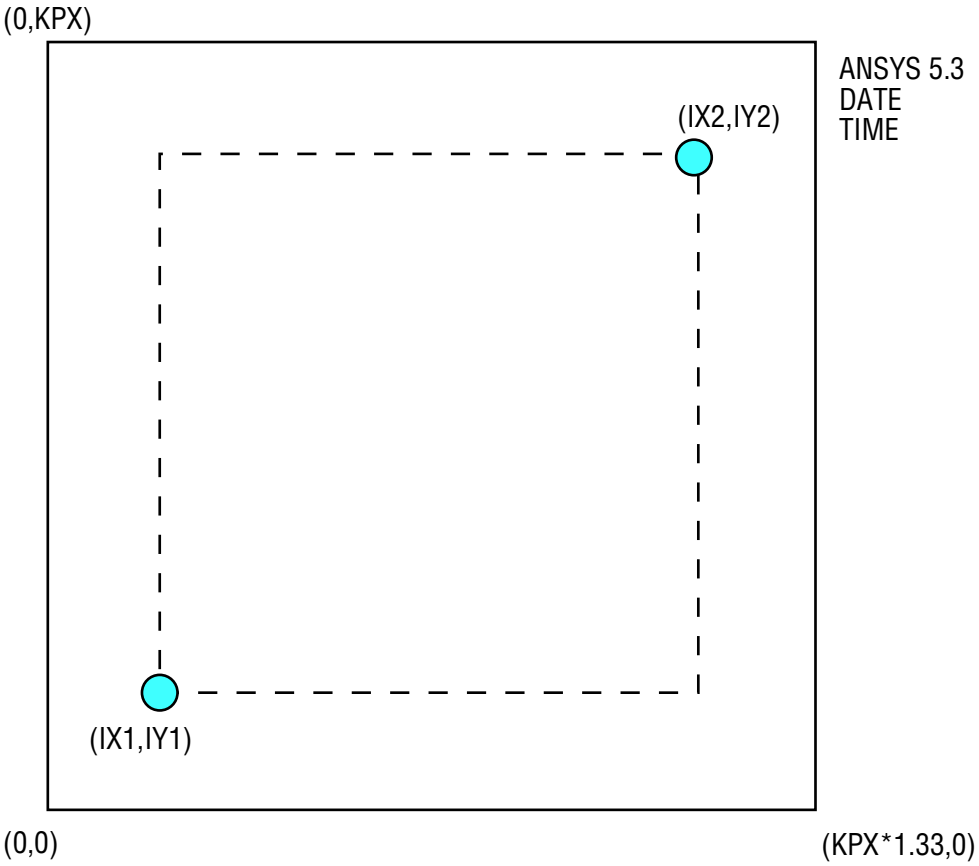
IX1,IY1 is the lower left corner of the z-buffer image.

IX2,IY2 is the upper right corner of the z-buffer image.

The image should be mapped to the hardcopy device accordingly.

The following graphic illustrates the items described above:

Figure 4.1 Display Format for Z-buffered Graphics



4.3. Neutral Graphics File Format

The neutral graphics file is an 80-byte, ASCII coded file with fixed length records. It contains plot directives representing the image of a display, as formed in ANSYS, encoded onto a host-independent, printable character set.

Most ANSYS users will not need to know the format of the graphics file. However, in rare cases, you may want to edit your graphics file or, as a programmer, you may need to know the file format to write a program that reads it. Although the file is ASCII coded, it can be difficult to interpret. This section gives details about the file format.

4.3.1. Characters the Graphics File Uses

The host-independent printable character set consists of the ASCII characters listed below:

- Numerals 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9
- Uppercase alphabetic characters A through Z
- The following characters: $ ( ) * + , - . < = >
- The space character, " ".

4.3.2. Graphics File Directives

Graphics files contain a set of directives that define various aspects of how ANSYS displays a plot, such as window coordinates, colors for graphs and test, line dimensions, and so on. Each directive consists of a command character followed by one or more parameters.

Within a graphics file, one directive directly follows the preceding directive. For example, below is the first line of a graphics file:

```
(BBAAA2A0AAAAAAPPPLPO>AP$MEKLKBAJANSYS 5.3$MEKLEFALNOV 15 1996$MEKKOJAI10:01:40
```

The text of this example line breaks down as follows:

`(BBAAA`	The Start-Plot directive, beginning with command character. (B, B, A, A, and A are the values of parameters defining the plot environment. (Parameters for all plot directives, and their possible values, are explained later.)
`2A`	The Text-Size directive, which determines the type size of displayed text strings. The 2 is the command character, and A represents the size value.
`0AAAAAAPPPLPO`	The Window directive, which sets the coordinates for the displayed image. 0 is the command character. AAAAAA represents the first set of coordinates (the lower left corner of the image), and PPPLPO represents the second coordinate set (the right upper corner of the image).
`>AP`	The Text-Color directive, which sets the color of displayed text. > is the command character. AP is a parameter value specifying the color.
`$MEKLKBAJANSYS 5.3`	The first of several Text directives. $ is the command character, MEKLKB are the coordinates for the text, AJ is the number of characters in the string, and ANSYS 5.3 is the text string itself.
`$MEKLEFALNOV 15 1996`	A second Text directive, defining the position and length of the string NOV 15 1996.
`$MEKKOJAI10:01:40`	A third Text directive, defining the position and length of the string 10:01:40

4.3.2.1. Parameter Types for Graphics File Directives

The descriptions of graphics file directives in the next section include discussions of the parameter or parameters for each directive. There are five types of parameters:

Parameter Type	Parameter Attributes	Valid Parameter Values
Int	1 byte, base 16 (letters A through P)	0 through 15
Long	2 bytes, base 16 (letters A through P)	0 through 255
Byt3	3 bytes, base 16 (letters A through P)	0 through 65535
Xy	6 bytes, base 16 (letters A through P)	0 through 4095, mapped to coordinate space of -1.0 to 1.67

String	An array of *Nchar* characters	Characters from the common character set.

4.3.2.2. Directive Descriptions

The next few pages describe each of the graphics file directives. Parameters are always specified in the order shown below.

Graphics Directive	Command Character	Parameters	Parameter Types
Start_Plot	(	keras - Defines whether the display surface is cleared prior to the plot (0 = do not clear the surface, 1 = clear it) kras - Defines whether the display uses raster mode or vector mode (1 = raster mode, 0 = vector mode) kcntr - Defines whether the display uses a contour color map or shading color map (1 = contour, 0 = shading) kdocu - Defines whether the Docu column is compressed (1 = do not compress, 0 = compress) ispare - A spare value	Int, Int, Int, Int, Int
Window	0	x1,y1, x2,y2 (x and y coordinates)	Xy, Xy
Area-Color	<	iclra - Sets the color for the displayed area. (See "Color Specification" below.)	Long
Graph-Color	=	iclrg - Sets the color for the displayed graph. (See "Color Specification" below.)	Long
Text-Color	>	iclrt - Sets the color for displayed text. (See "Color Specification" below.)	Long
Text-Size	2	tsize - Defines the size of displayed text (0 = normal, 1 = small)	Int
Line-Type	,	ltype - Defines the type of lines used in the display (0 = solid, 1 = dashed)	Int
Line-Width	1	lwidth - Defines the width of displayed lines (0 = normal, 1 to 5 = larger line size)	Int
Marker Size	3	size - Defines the size of the node marker (0 = the smallest size, 15 = the largest size)	int
Point	*	x,y - Defines a point at coordinates x,y	Xy
Move	.	x,y - Moves to coordinates x,y	Xy
Draw	-	x,y - Draws a line to coordinates x,y	Xy
Text	$	x,y - Sets coordinates for where text will display nchar - Defines the number of displayed characters string - Defines the text string itself	Xy, Long, String
Normal	/	inorm - This value, divided by 255, is cos(), where is the viewing direction and the surface normal of subsequent polygons	Long

Graphics Directive	Command Character	Parameters	Parameter Types
Polygon	+	n - Defines the number of polygon vertices kedge - Defines whether the polygon edge is displayed (0, = do not display edge, 1 = display it) xy1,...xyn - Defines coordinates for the polygon	Int, Int, Xy...Xy
No-Op	none	no parameters	none
End-Plot	)	no parameters	none
Pixmap	Z	kpx - Defines the pixel resolution x1,y1 - Sets coordinates for the lower left corner of the z-buffer image x2,y2 - Sets coordinates for the upper right corner of the z-buffer image The following three parameters are run-length encoded data which repeats until all pixels are read in, as defined by the window (X2-X1 + 1) * (Y2-Y1 + 1) n - Defines the number of pixels in a row inrm - Sets the normal for pixels iclr - Sets the color for the pixmap	Byt3, Xy, Xy, Long, Long, Long, ...

4.3.2.3. Color Specification

Below is the list of color specifications used by the directives that set colors for areas, graphs, and text. If more than a single intensity of a color is available, use the value specified by the **Normal** directive to complete the selection. *Normal* of 0 represents the lowest intensity and *normal* of 255 represents the highest intensity.

Value	Color
0	Black
1	Cyan
2	Blue-Magenta
3	Red
4	Cyan-Blue
5	Magenta-Red
6	Green
7	Orange
8	Magenta
9	Yellow-Green
10	Blue
11	Green-Cyan
12	Yellow
13	Dark Gray
14	Light Gray
15	White

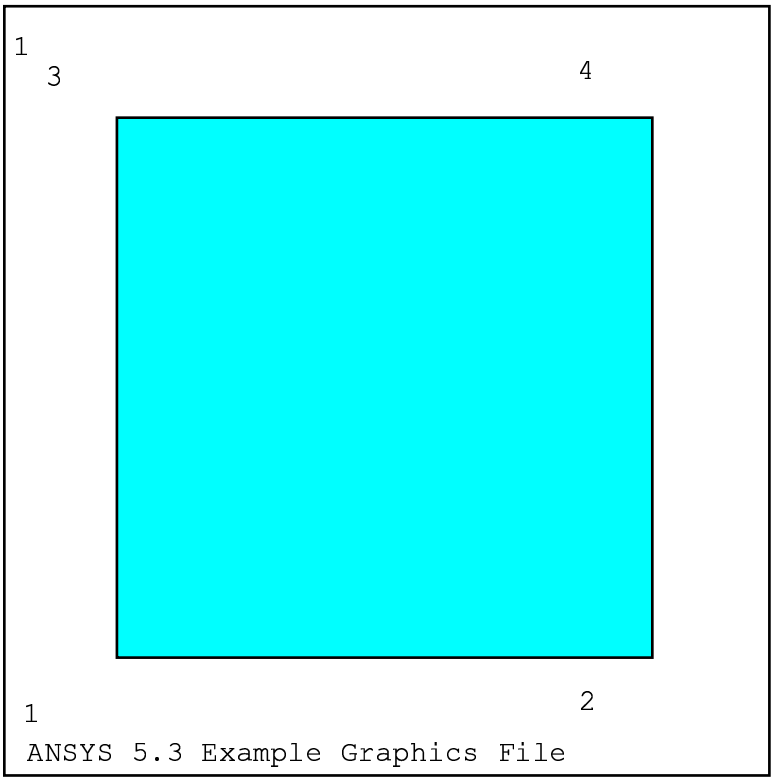
Value	Color
16	
:	Reserved for future use
127	
128	Blue
:	:
	Cyan
	:
	Green
	:
	Indices 128 through 255 represent the color spectrum used to display the Low (Blue) to High (Red) contour values.
	:
	Yellow
	:
	Orange
:	:
255	Red

4.4. Decoding a Graphics File: an Example

The following example shows you the following:

- The ANSYS command stream used to create a simple graphics plot, shown in Figure 4.2: "Example Display of a Graphics File" below
- The encoded graphics file that these commands produce
- The decoded graphics plot directives

Figure 4.2 Example Display of a Graphics File



4.4.1. The Example Command Stream

To create the graphics display shown in Figure 4.2: “Example Display of a Graphics File”, you would issue the following ANSYS commands:

```
/PREP7
/TITLE, ANSYS 5.3 Example Graphics File
N,1,1
N,2,2
NGEN,2,2,1,2,1,,1
ET,1,42
E,1,2,4,3
/PNUM,ELEM,1
/PNUM,NODE,1
/SHR,.1
/SHOW,F33
EPLOT
FINISH
```

4.4.2. Example Graphics File Contents

The commands listed above produce the display shown in Figure 4.2: “Example Display of a Graphics File” and the following graphics file:

```
(BBAAA2A0AAAAAAPPPLPO&#60AA&#62AP$MEKLKBAJANSYS 5.3$MEKLEFALNOV 16 1996$MEK
KOJAI15:57:07$MEKKIMAMPLOT NO.   1$MEKKDAAIELEMENTS$MEKJNEAIELEM NUM2
B0AAAAAALPOLPO&#60AB/PP+EBBBHBBHKOGBBHKOGKOGBBHKOG$FPPFPPAB1$AILAILAB1$L
HCAILAB2$LHCLHCAB4$AILLHCAB32A0AAAAAAPPPLPO.AAAAAA-LPOAAA-LPOLPO-AAAL
PO-AAAAAA>AB$ABLLKBAB1>AP$MEKJBLAGZV  =1$MEKILPAJDIST=0.55$MEKIGCAIXF
  =1.5$MEKIAGAIYF  =0.5$MEKHKKAPCENTROID HIDDEN$ABOABOCA ANSYS 5.3 Ex
ample Graphics File)
```

The decoded plot directives are:

(BBAAA	Start-Plot: /ERASE, raster mode
2A	Text-Size: Default
0AAAAAAPPPLPO	Window: 0.0 4095,3070
&#60AA	Area-Color: Black
&#62AP	Text-Color: White
$MEKLKBAJANSYS 5.3	Text: 3146 2977 "ANSYS 5.3"
$MEKLEFALNOV 16 1996	Text: 3146 2885 "NOV 15 1996"
$MEKKOJAI15:57:07	Text: 3146 2793 "15:57:07"
$MEKKIMAMPLOT NO. 1	Text: 3146 2700 "PLOT NO. 1"
$MEKKDAAIELEMENTS	Text: 3146 2608 "ELEMENTS"
$MEKJNEAIELEM NUM	Text: 3146 2516 "ELEM NUM"
2B	Text-Size: Small
0AAAAAALPOLPO	Window: 0 0 3070 3070
&#60AB	Area-Color: Cyan
/PP	Normal: 255
+EBBBHBBHKOGBBHKOGKOGBBHKOG	Polygon: 279, 279, 2790, 279 2790, 2790 279, 2790
$FPPFPPAB1	Text: 1535 1535 "1"
$AILAILAB1	Text: 139 139 "1"
$LHCAILAB2	Text: 2930 139 "2"
$LHCLHCAB4	Text: 2930 2930 "4"
$AILLHCAB3	Text: 139 2930 "3"
2A	Text-Size: Default
0AAAAAAPPPLPO	Window: 0,0 4095,3070
.AAAAAA	Move: 0,0
-LPOAAA	Draw: 3070,0
-LPOLPO	Draw: 3070,3070
-AAALPO	Draw: 0,3070
-AAAAAA	Draw: 0,0
>AB	Text-Color: Cyan
$ABLLKBAB1	Text: 27 2977 "1"
>AP	Text Color: White

`$MEKJBLAGZV  =1`	Text: 3146 2331 "ZV =1"
`$MEKILPAJDIST=0.55`	Text: 3146 2239 "DIST=0.55"
`$MEKIGCAIXF=1.5`	Text: 3146 2146 "XF =1.5"
`$MEKIAGAIYF  =0.5`	Text: 3146 2054 "YF =0.5"
`$MEKHKKAPCENTROID` ` HIDDEN`	Text: 3146 1962 "CENTROID HIDDEN"
`$ABOABOCA ANSYS 5.3` ` Example Graphics File`	Text: 30 30 "ANSYS 5.3 Example Graphics File"
`)`	End-Plot
	No-Op

Index

I

J

L

M

N

P

R

S

T

W

Z