The Focus PADT
A Publication for ANSYS Users

Contents

Feature Articles

On the Web

Resources

© 2002, by Phoenix Analysis & Design Technologies, Inc. All rights reserved.

The Focus PADT
A Publication for ANSYS Users

Tools of Post Processing: Listing

By Eric Miller, PADT

Post processing is an oft-maligned part of the analysis process. Real analysts focus on meshing and running their models. But the fact of the matter is that getting *meaningful answers* is the whole point of doing a simulation, so spending a little bit of time to learn how to produce more concise and digestible results is a good investment. To that end, we will be covering a couple of key ways to produce higher quality results using POST1 and POST26, including plotting, animation, sections, and for this first article, listing.

Listing Results

Because of ANSYS' long history, it has a rich listing capability. Back in the old days, listing was the primary way to extract information. But even now, it is still an important way to nail down facts about your results. By default, the formatting is not very clean and it is easy to get buried in details. Learning a few commands can turn that pile of numbers into valuable tables and eliminate considerable manual of cleanup in external programs.

The first thing to know about ANSYS listings is that they are set up for sending data directly to large volume line printers used 10 or 15 years ago. That's why that silly header shows up every 25 lines. It's is also the reason why the information is presented in unformatted Courier1 font without tabs, separating lines, bolding, or italics. Old printers just could not produce that type of formatting. So don't try to dress up listings by changing font or style unless you insert tabs into your data first.

The most important formatting command is (*surprise!*), /format:

/FORMAT, NDIGIT , *Ftype*, *NWIDTH*, *DSIGNF*, *LINE*, *CHAR*>

- *NDIGIT*: Number of digits in first column (usually entity number)
- *Ftype*: FORTRAN format type to use: G, F, or E
- *NWIDTH*: Total width of number
- *DSIGNF*: Significant figures
- *LINE*: Number of lines per page (defaults to /page)
- *CHAR*: Number of characters per line before wrap (defaults to /page)

Figure 1 shows listings with the default formatting and with controls placed on spacing. The listing wraps and is hard to read with the defaults (formatted for a 132 column line-printer.) You could also use the *DSIGNF* argument if you need more precision on a number.

The Focus PADT
A Publication for ANSYS Users

Default /format (Windows)

PRINT S NODAL SOLUTION PER NODE

NODE	SX	SY	SZ	SXY	SYZ	SXZ	S1	S2	S3	SINT	SEQV
1	5.6829	295.48	-35.332	31.429	-29.368	8.1254E-03	291.38	5.2283	-36.455	333.83	304.43
2	-35.683	399.89	-29.829	-6.9341	-28.458	-34.454	488.32	2.8228	-61.695	461.82	431.35
3	4.5651	265.62	-1.8636	36.483	38.882	3.3833	269.98	5.1495	-33.328	283.23	214.28
4	-35.448	218.38	-368.83	-35.888	38.358	3.1198	213.14	-36.355	-368.95	432.69	383.49

/format,8,f,9,1,100,240

PRINT S NODAL SOLUTION PER NODE

NODE	SX	SY	SZ	SXY	SYZ	SXZ	S1	S2	S3	SINT	SEQV
1	5.1	295.4	-35.3	31.4	-29.2	8.1	291.2	5.2	-36.5	333.6	304.4
2	-35.6	399.1	-29.8	-6.9	-28.4	-34.5	488.3	2.8	-61.7	461.8	431.3
3	4.6	265.6	-1.9	36.4	38.8	3.2	269.9	5.1	-33.3	283.2	214.2
4	-35.4	218.3	-368.8	-35.8	38.3	3.8	213.1	-36.2	-383.8	432.1	383.5
5	-35.1	-345.3	51.4	-2.3	-6.8	-82.9	333.4	-69.6	-345.2	456.6	398.3
6	-4.8	-369.8	238.2	-2.6	-9.2	54.5	253.8	-31.5	-369.2	688.2	522.3
7	33.3	634.8	-85.8	3.4	1.8	31.5	634.9	34.4	-89.8	183.9	658.4

Figure 1.

The next area that needs user-tweaking is cleaning up the header. The /HEADER command contains a series of on/off switches that can be used to simplify and reduce the amount of information printed at the top of data listings. It is also very useful when you want to dump a listing to a file that will be read by another program. This is because you can turn off that pesky header information that is such a pain to skip over. The command is fairly simple:

/HEADER, *Header*, *Stitle*, *Ldstmp*, *Notes*, *Colhed*, *Minmax*

- *Header*: Standard ANSYS header (batch only, always off in interactive mode)
- *Stitle*: Subtitles
- *Ldstmp*: Load step information for listed data
- *Notes*: Information on a particular table listing
- *Colhed*: Column headings on table listings
- *Minmax*: Min/Max information at the end of tables

Figure 2 shows the default header with all that extra junk and spaces, as well as two alternative listings that are much more readable.

The Focus PADT
A Publication for ANSYS Users

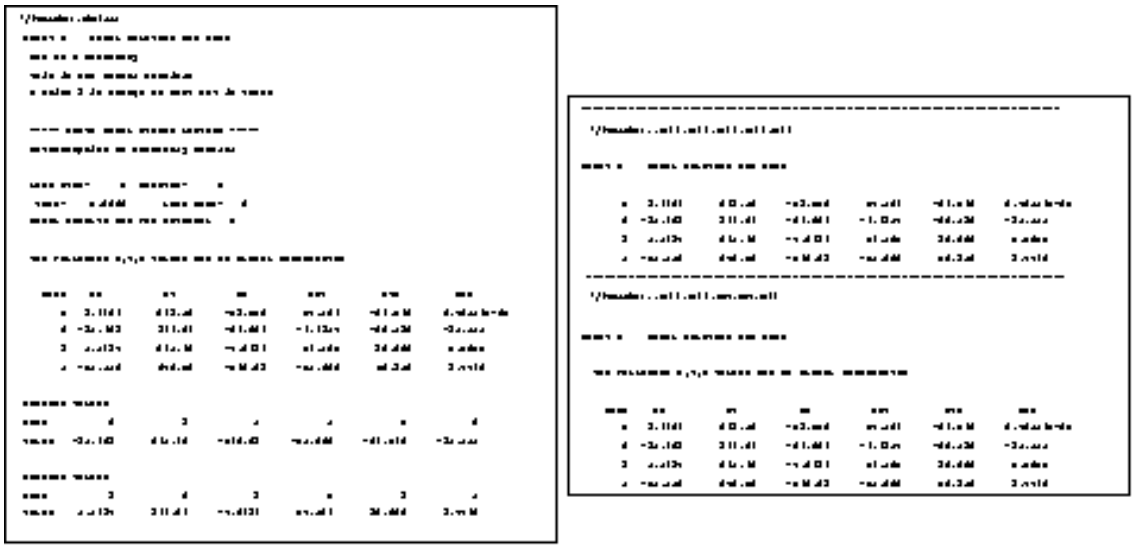


Figure 2.

However, this does not solve the most annoying aspect of saving listings when running in batch mode to a file – a header every 24 lines. You control this with the /PAGE command:

/PAGE, *ILINE*, *ICHAR*, *BLINE*, *BCHAR*

- *ILINE*, *ICHAR*: no longer used
- *BLINE*: Number of lines per page for batch, /output or interactive listing
- *BCHAR*: Number of characters per line for batch, /output or interactive listing

I set *BLINE* to 9999 and get only one header.

But what if ANSYS does not list the data you want, or does not list it in a way you find useful. Since you can extract information from the database with *GET's and *VGET's you can pretty much format information any way you want. The most common is to do a *VGET followed by *CFOPEN, *VWRITE's, and *CFCLOSE. *VWRITE allows you to enter FORTRAN and C format statements. There is also an *MWRITE command for outputting matrices. With these, output to programs like Microsoft Excel is a breeze. Figure 3 shows a very simple macro that outputs node numbers and X locations for 10 nodes using various formats.

```
finish
/clear
/prep7
n,1
n,10,1
fill,1,10

*dim,nnx,,10,2
*vfill,nnx(1,1),ramp,1,1
*vget,nnx(1,2),node,1,loc,x

*cfopen,vwrt,txt
```

The Focus PADT
A Publication for ANSYS Users

```
*vwrite,
----- Comma Separated 'C':
*vwrite,
(" ")
*vwrite,nnx(1,1),nnx(1,2)
%8d,%16.9g

*vwrite,
(" ")
*vwrite,
----- Comma Separated FORTRAN:
*vwrite,
(" ")
*vwrite,nnx(1,1),nnx(1,2)
(F8.0,",",G16.9)

*vwrite,
(" ")
*vwrite,
----- Text in the line:
*vwrite,
(" ")
*vwrite,nnx(1,1),nnx(1,2)
Node: %8d X-Position: %16.9g

*vwrite,
(" ")
*vwrite,
----- FORTRAN Again
*vwrite,
(" ")
*vwrite,nnx(1,1),nnx(1,2)
(" -- ",F8.0,": ",E16.9)
*cfclose
*uilist,vwrt,txt
```

Figure 3. Example macro.

```
----- Comma Separated 'C':

   1,  0.000000000
   2,  0.250000000
   3,  0.500000000
   4,  0.750000000
   5,  1.000000000
```

The Focus PADT
A Publication for ANSYS Users

----- Comma Separated FORTRAN:

```
1.,    0.000000000
2.,    0.250000000
3.,    0.500000000
4.,    0.750000000
5.,    1.000000000
```

----- Text in the line:

```
Node:      1 X-Position:    0.000000000
Node:      2 X-Position:    0.250000000
Node:      3 X-Position:    0.500000000
Node:      4 X-Position:    0.750000000
Node:      5 X-Position:    1.000000000
```

----- FORTRAN Again

```
--      1:  0.000000000E+00
--      2:  0.250000000E+00
--      3:  0.500000000E+00
--      4:  0.750000000E+00
--      5:  0.100000000E+01
```

Figure 4. Output in vwrt.txt.

Take note that the first example is comma separated. A *VWRITE or *MWRITE with commas has proven to be the best way to output to Microsoft Excel. I prefer the C formatting because it is more concise and allows for integers. For various reasons, the FORTRAN (I) format does not work well on some platforms.

However, if that is not enough control and you want to include fancy formatting and your own text headers, the best solution is to write an external command (~command) or use the built in Tcl/Tk interpreter. These options allow you to extract information with *VGET's, and then output them with the full formatting provided by a modern programming language.

In short, you can pretty much format output any way you want with a little bit of set up, especially with the *VWRITE command. I have found this especially useful when I am automating report generation or when I'm doing a parametric study. Scripting the format of the listing allows me to have consistent and readable output. Some say the days of six-foot high stacks of result printouts are gone, but that does not mean that you cannot find value in outputting values.

---

1 Courier belongs to a family of fonts called "monotype" or "fixed width". What this means is that each character is exactly the same width. Most fonts are "proportional", which means they

The Focus PADT
A Publication for ANSYS Users

have characters that vary in width for aesthetic reasons, which makes formatting without tabs a real pain. To illustrate, here are two examples of some lines of text The first pair is in Times New Roman (a proportional font) and the second is in Courier. Compare the ends of the lines.

This line has exactly forty characters!!
Hello, I am forty characters long. Even?

This line has exactly forty characters!!
Hello, I am forty characters long. Even?

The Focus PADT
A Publication for ANSYS Users

User Programmable Features

by Eric Miller, PADT, and Rod Scholl, PADT

UPFs: What Are They?

One of ANSYS' strengths is that its toolbox will handle anything from a textbook plate with hole, all the way up to allowing you to recompile the code with FORTRAN scripts that you develop yourself. The User Programmable Feature (UPF) capability allows users to create their own subroutines and link them into ANSYS.

What Are UPFs Used For?

There are a variety of common uses for UPFs:

- Material models
- User-developed elements
- User commands
- Element orientation
- User beam definition
- Heat generation
- Defining real constants "on the fly"

How Hard Is It to Implement a UPF?

Implementing a UPF is definitely more work than simply programming the action using APDL. However, if you have existing FORTRAN code you want to include or run externally, or have any of the above-mentioned challenges, then APDL will not suffice.

To learn for yourself, you will want to surround your self with resources:

- This article!

The Focus PADT
A Publication for ANSYS Users

- *ANSYS Programmer's Guide: User Programmable Features*
  - A **must-have**!!!!
  - All the needed subroutines are documented here
- The hundreds of documented template scripts found within the installation directory in /custom/user
- XANSYS. Submit your questions to the XANSYS community.
- PADT. Submit your questions to us.

What is the Process?

1. Copy a sample routine from the ANSYS directories
2. Modify it to do what you want
3. Use the ANSYS batch compile script to compile and link
4. Locate the new executable for use
5. In your model, activate user features by applicable method
   a. Define user element
   b. Define user material model
   c. Turn on user feature in element real constants
   d. Define user command with /UCMD
   e. Setup in USRCAL command

6. Debug

Do I have to use FORTRAN?

- All of these routines are called as FORTRAN and use FORTRAN calls to utility functions.
- You could use FORTRAN wrappers to call C routines, but that is usually not worth the effort.

The Focus PADT
A Publication for ANSYS Users

- ANSYS provides you with a compile script to help with the process.
- Don't freak out. FORTRAN is very easy to use, and creating UPFs and recompiling ANSYS is actually very easy.

You Want to Learn by Example?!?

So before opening the *ANSYS Programmer's Guide*, you want to take a look at what you're getting into? Well then, buckle up...

UPFs Example1, USER02.F

- USER02.F contains code to offset nodes by supplied x,y,z values
- Good example of structure of routines
- Shows how other routines are called

Typical Header Information. Includes source control data (*deck), copyright information, confidentiality, and pointer to more information. Also function statement.

```
*deck,user02          user
ANSYS,INC
     function  user02 (initin,dpin,ch4in,ch8in)
c *** primary function:   user routine number 02
c     This demonstration offsets selected nodes with the command:
c       usr2,dx,dy,dz

c     *** ansys (r) copyright(c) 2000
c     *** ansys, inc.
c *** Notice - This file contains ANSYS Confidential information

c /*************************************************************\
c | see user01 for additional information on user routines      |
c \*************************************************************/
<snip>
```

Arguments Section. Describes input and output arguments, what type they are, and what they are used for. Also, additional examples and help are put in this section. Externally passed and local variables are also defined here. Finally, the standard include statements are placed here.

The Focus PADT
A Publication for ANSYS Users

```
<snip>
c input arguments:
c   variable (typ,siz,intent)   description
c   intin    (int,ar(12),in)  - integers from command line
c   dpin     (dp,ar(12),in)   - double precision from cmnd line
c   ch4in    (ch*4,ar(12),in) - upper case 4 characters from cmnd
c   ch8in    (ch*8,ar(12),in) - as input 8 characters from cmnd

c output arguments: none
c   user02   (int,sc,out)     - result code (should be zero)
c                               (which is ignored for now)

c ***************************************************************
c  intin(2), dpin(2), ch4in(2), & ch8in are all representations
c  of the contents of the second field on the command line
c  (the field after the command label)
c ***************************************************************
      external  wrinqr,ndinqr,ndgxyz,ndpxyz,erhandler
      integer   wrinqr,ndinqr,ndgxyz

      integer  usr02,intin(12),iott,maxnp,i,ksel
      double precision  dpin(12),xyz(3)
      character*4  ch4in(12)
      character*8  ch8in(12)

#include "ansysdef.inc"
<snip>
```

Calculations. Use ndinqr() to get max node number. This code snippet loops on all possible node numbers, checks to see if the node is selected, and gets its *x*, *y*, and *z* values. Then it does the offset and updates the *x*,*y*,*z* position with ndpxyz().

```
<snip>
      maxnp = ndinqr (0,DB_MAXDEFINED)

      do i = 1,maxnp
         ksel = ndgxyz (i,xyz(i))
         if (ksel .eq. 1) then
            xyz(1) = xyz(1) + dpin(2)
```

The Focus PADT
A Publication for ANSYS Users

```
            xyz(2) = xyz(2) + dpin(3)
            xyz(3) = xyz(3) + dpin(4)
            call ndpxyz (i,xyz(i))
         endif
      enddo
<snip>
```

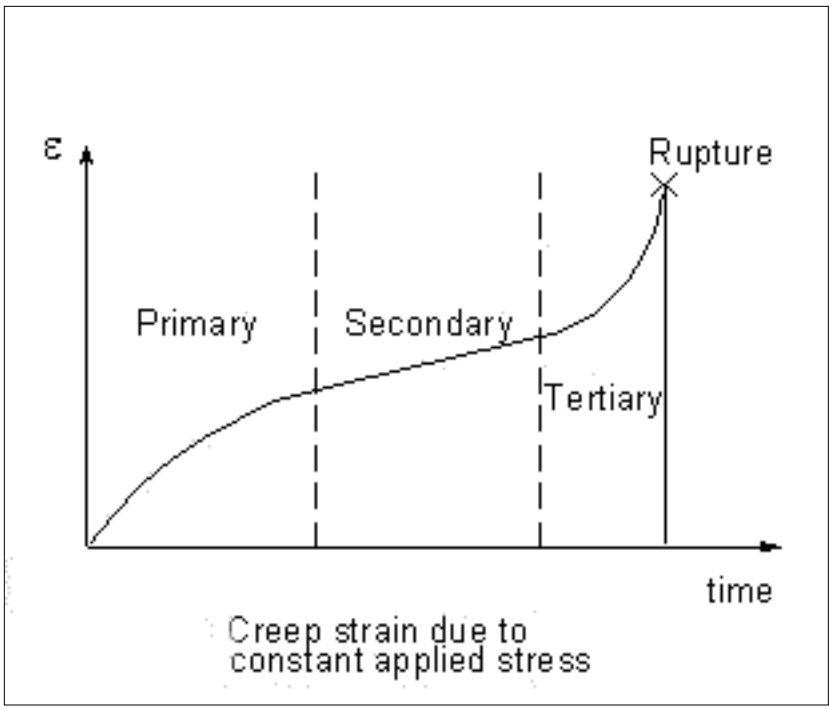
Clean Up and Exit. This code snippet is used to figure out the standard ouput unit, and then write a message to it. It will also write a message to a GUI pop-up box. It will then set the return value and leave.

```
<snip>
c ***** write to output file *****
      iott = wrinqr(WR_OUTPUT)
      write (iott,2000)
 2000 format (/' NODE OFFSET COMPLETE '/)

c ***** write to GUI window *****
      call erhandler ('user02',3000,
     x     2,'NODE OFFSET COMPLETE',0.0d0,' ')

c ***** required return value *****
      user02 = 0

      return
      end
<snip>
```

UPFs: Example 2, USERCR.F

Creep strain due to constant applied stress:

The Focus PADT
A Publication for ANSYS Users



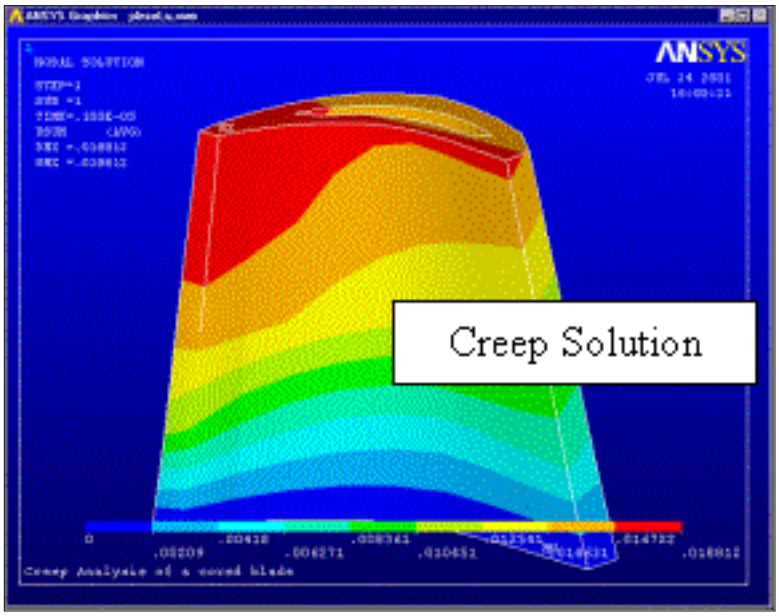

```
subroutine usercr (elem,intpt,mat,ncomp,kfirst,kfsteq,e,posn,d,
     x proptb,timval,timinc,tem,dtem,toffst,fluen,dfluen,epel,
     x epcrp,statev,usvr,delcr)
c
#include "impcom.inc"
      external erhandler
c
c user-defined fortran parameters
c     --- size of usvr data
      integer nuval,nintp
      parameter (nuval=1,nintp=1)
c
c external subroutines and functions
      external egen
      double precision egen
      external vapb1,vamb1,vmult
c
c integer variables
      integer elem,intpt,mat,ncomp,kfirst,kfsteq
c
c double precision variables
      double precision
```

The Focus PADT
A Publication for ANSYS Users

```
     x e,posn,d(ncomp,ncomp),proptb(72),timval,timinc,tem,dtem,
     x toffst,fluen,dfluen,epel(ncomp),epcrp(ncomp),
     x statev(ncomp*5+2),usvr(nuval,nintp),delcr,temabs,con,
     x del(6),epet,sigen,eptot,ept(6),
c -PADT--- PADT added variables
     x qovrk,c1,alpha,dbln,crprt,vrbflg,
     x aaaa,bbbb,cccc,dddd
<snip>

<snip>
c
c --- initial checks
c --- author should remove the warning below when making changes
      if (intpt.eq.1 .and. kfirst.eq.1)
c
     x call erhandler('usercr',5000,2,
     x  'ANSYS,INC.-supplied version of coding for USERCR
     x  has been used.'
     x  ,0.0d0,' ')
c
      if (nuval*nintp.gt.840)
     x call erhandler('usercr',5010,4,
     x  'Maximum storage allowed by USVR has been exceeded.'
     x  ,0.0d0,' ')
c
c --- initialize creep strain increment in case on creep
      delcr = 0.0d0
c --- no creep if time is not moving
      if (timinc .le. 0.0d0 .and timval .le. 0.0d0) fo to 999
c --- no creep if temperature is not defined
      temabs = tem + toffst
      if (temabs .le. 0.0d0) then
c
      call erhandler('usercr',5020,3,
     x 'Temperature= %G must be greater than zero for creep.'
     x ,temabs,' '_
c
      go to 999
      endif
```

The Focus PADT
A Publication for ANSYS Users

<snip>



<snip>

```
c
c ***** define the equivalent strain and stress *****
c --- define the equivalent strain using the function egen
      epet = egen (nocomp,epel(1),posn)
c --- no cree if the strain is zero
      if (epet .eq. 0.0d0) go to 999
c --- define the stress
      sigen = e*epet
c
c ***** define the creep strain rate *****
c ***** normal beginning of user changes
c
c -PADT--- Put table values in constants for clarity
c
      c1 = proptb(1)
      alpha = proptb(2)
      dbin = proptb(3)
      vrbflg = proptb(5)
c
c -PADT--- If table item 4 (Q/k) is 0, set default of 8110
      if (proptb(4) .eq. 0) then
```

The Focus PADT
A Publication for ANSYS Users

```
        qovrk = 8110
      else
        qovrk = proptb(4)
      endif
<snip>

<snip>
c
c --- define the creep strain rate

      crprt = c1*(sinh(alpha*sigen)**dbln)*dexp(-1*qovrk/temabs)

c ***** end of sample creep law
c ***** normal end of user changes
c
c ***** compute creep strain increments for each component *****
c
c --- define the creep strain increment from the rate
      delcr = crprt*timinc
c
c --- use prandtl-reuss relations to compute increments for each
      if (ncomp .eq. 1) then
        del(1) = delcr*epel(1)/epet
      else
        con = delcr/epet
        con = con/(2.0d0*(1.0d0+posn))
        call vmult (epel(4),del(4),ncomp-3,3.0d0*con)
        del(1) = con*(2.0d0*epel(1) - epel(2) - epel(3))
        del(2) = con*(2.0d0*epel(2) - epel(3) - epel(1))
        del(3) = con*(2.0d0*epel(3) - epel(1) - epel(2))
      endif
c
c --- update the strains
      call vapb1 (epcrp(1),del(1),ncomp)
      call vamb1 (epel(1),del(1),ncomp)
<snip>

<snip>
c -PADT--- Do Verbose I/O, if vrbflg eq 1
      if (intpt.eq.1 .and. vrbflg .eq. 1) then
```

The Focus PADT
A Publication for ANSYS Users

```
      iott=6
      write (iott,*) 'element number = ',elem,' gauss point 1'
      write (iott,1) 'time=',timval,' stress=',sigen,' temp=',tem
      write (iott,2) 'delcr=',delcr,' del=',del(1)
      endif
c -PADT--- Do Summary I/O, if vrbflr eq 2
      if (intpt.eq.1 .and. vrbflg .eq. 2) then
      iott = 34
      if (elem .eq. 1) then
      epet = egen (ncomp,epel(1),posn)
      write (iott,3) elem,timval,sigen,temabs,crprt,delcr,epet
      endif
      endif
c
 999  return
 1    format(2x,3(a,d15.8))
 2    format(2x,2(a,d15.8))
 3    format(2x,i6,6(',',g16.9))
      end
```

ANSYS Input File for USERCR

```
!ANSYS Input File for USERCR
! The ANSYS TB values are as follows:
! 1 C1
! 2 alpha-sub-1
! 3 n
! 4 Q/k (defaults to 8110)
! 5 Verbose output flag (1 = verbose output, 0 = standard output)
!
c1 = 9.62e4
a1 = 6e-4
n1 = 3.3
qovrk = 8110
vrbs = 2
TB,CREEP, 1
tbdata,6,100 !use user creep
tbdata,1,c1
tbdata,2,a1
```

The Focus PADT
A Publication for ANSYS Users

```
tbdata,3,n1
tbdata,4,qovrk
tbdata,5,vrbs
```

UPFs: Lessons to Consider

- Always try APDL macros first
- Consider external commands to avoid custom versions of ANSYS
- You must have the proper FORTRAN compiler
  - See installation guide
- KISS
- Start by replicating the existing ANSYS functionality
  - Add your changes slowly
- Write and debug on simple models
- Don’t be afraid of write(*,*)

The Focus PADT
A Publication for ANSYS Users

FEA for .EDU

by Rod Scholl, PADT

Finite Element Analysis (FEA) using a commercial code is becoming a more common skill for undergraduate mechanical engineers. Perhaps PADT can help you at your .edu institution by providing free course materials — and maybe more! The course material exchange program pools the resources of professors nationwide into an evolving curriculum that you can take advantage of. This and other free benefits may be available to you.

Please see the recent ANSYS press release for more information.

“They get younger every year!”

Drop us an email and we can discuss how we might help!

The Focus PADT
A Publication for ANSYS Users

Upcoming Training at PADT

A complete schedule of the training courses for the next three months is shown below. Learn more about how the **Training Services** offered by PADT can save you time and money. Or, feel free to drop an e-mail to our training coordinator, Ted Harris!

Month	Start	End	Course	Description	Location
Apr '04	4/5	4/7	101	Introduction to ANSYS, Part I	Irvine, CA
	4/19	4/21	101	Introduction to ANSYS, Part I	Albuquerque, NM
	4/21	4/23	201	Basic Structural Nonlinearities	Tempe, AZ
May '04	5/3	5/5	101	Introduction to ANSYS, Part I	Tempe, AZ
	5/6	5/7	102	Introduction to ANSYS, Part II	Tempe, AZ
	5/10	5/11	100	Engineering with Finite Element Analysis	Tempe, AZ
	5/13	5/14	803	Tcl/Tk for ANSYS	Tempe, AZ
	5/19	5/20	203	Dynamics	Tempe, AZ
Jun '04	6/7	6/9	101	Introduction to ANSYS, Part I	Irvine, CA
	6/21	6/23	201	Basic Structural Nonlinearities	Tempe, AZ
	6/24	6/25	202	Advanced Structural Nonlinearities	Tempe, AZ
	6/28	6/29	301	Heat Transfer	Irvine, CA

Whether it s one of our regularly-scheduled classes at our training facility, or a customized class tailored to your specifications and location, you can be assured that the training you receive will have immediate, positive results on your capabilities in design, analysis, and product and process improvement.

A Publication for ANSYS Users

About *The Focus*

The Focus is a periodic electronic publication published by PADT, aimed at the general ANSYS user. The goal of the feature articles is to inform users of the capabilities ANSYS offers and to provide useful tips and hints on using these products more effectively. *The Focus* may be freely redistributed in its entirety. For administrative questions, please contact Rod Scholl at PADT.

The Focus Library

All past issues of *The Focus* are maintained in an online library, which can be searched in a variety of different ways.

Contributor Information

Please don t hesitate to send in a contribution! Articles and information helpful to ANSYS users are very much welcomed and appreciated. We encourage you to send your contributions via e-mail to Rod Scholl.

Subscribe / Unsubscribe

To subscribe to or unsubscribe from *The Focus*, please visit the PADT e-Publication subscriptions management page.

Legal Disclaimer

Phoenix Analysis and Design Technologies (PADT) makes no representations about the suitability of the information contained in these documents and related graphics for any purpose. All such document and related graphics are provided as is without warranty of any kind and are subject to change without notice. The entire risk arising out of their use remains with the recipient. In no event, including inaccurate information, shall PADT be liable for any direct, consequential, incidental, special, punitive or other damages whatsoever (including without limitation, damages for loss of business information), even if PADT has been advised of the possibility of such damages.

The views expressed in *The Focus* are solely those of PADT and are not necessarily those of ANSYS, Inc.