

The Focus



A Publication for ANSYS Users

Contents

Feature Articles

- [Slicing Your Mesh – The PSMESH Method](#)
- [C Descriptors, C Descriptors Run](#)
- [Improved Meshtool GUI](#)

On the Web

- [APDL Guide available for purchase](#)
- [Material Database for Workbench](#)
- [PADT seeks new analyst to join our team](#)



Resources

- [PADT Support: How can we help?](#)
- [Upcoming Training at PADT](#)
- [About *The Focus*](#)
 - *The Focus* Library
 - Contributor Information
 - Subscribe / Unsubscribe
 - Legal Disclaimer

ANSYS
Support Distributor

The Focus



A Publication for ANSYS Users

Slicing Your Mesh – The PSMESH Method

by Rod Scholl, PADT

Ever NUMMRG some nodes together, then later regret it, but it's too far back to rewrite your scripts or recover the log file? How about simply wanting to split a 3-D mesh into two? Ever want to insert contact behavior at a glued solid surface where the mesh must share nodes? Then you might wish you could make a boolean slice to your mesh, independent of any solid modeling.



There are some tricks to do these things by copying a new coincident set, then deleting the old. This may pose some numbering problems for you, but perhaps you can live with that. However, occasionally, you might find a connectivity issue: you can't move/copy the elements because they are associated with solids (and you don't want to undergo a MODMSH,DETACH and its disadvantages). You can't move/copy the solids because they are meshed! In fact, you would have to clear the mesh on all the glued adjacent solids too, *then* copy/delete, then re-glue the desired boundaries, then remesh them all.



Once in a while we find ourselves worked into a corner (okay, I guess I often work myself into a corner), and the thing we need is to divide a mesh into two parts that don't share nodes and just ignore the solid model stuff. That is where the PRET179's PSMESH command comes in.

Want to know more about implementing pretension with the PRET179s?

The first time I used the pretension joints I found they have some advantages over using thermal expansion or bonded contact offsets (e.g. CNOF) to implement prestrain in a model. The ANSYS help sections are pretty good on this. But so as not to disappoint, attached is a [script](#) of a nice implementation of multiple pretensions. It implements successive tightening of a wheel's spokes in a specified order and force. Good luck.



One of the internal steps in implementing pretension with the 179 elements is that ANSYS will divide the mesh for you at a specified node! I don't know what algorithm is used internally, but I with a little help from Sam, I'm a'gonna pirate it.



The procedure is to orient the current coordinate system normal to the plane you want

The Focus



A Publication for ANSYS Users

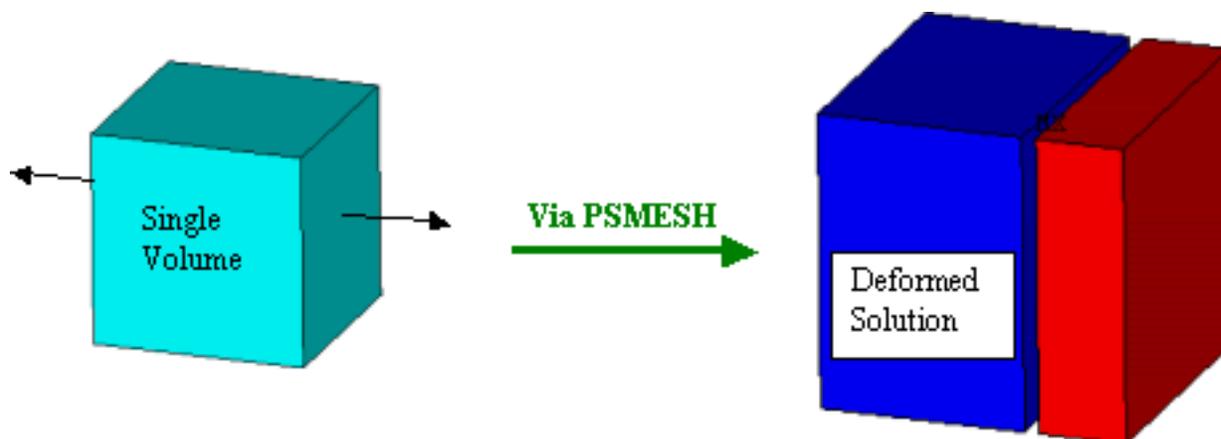
to cut at (doesn't matter if it's the X, Y, or Z axis) and then figure out a node (or location) where the cut should be made. Then just issue the PSMESH command and do a little cleanup!



For the following example, we wish to cut it at the $\frac{3}{4}$ -point (0.75,0,0), without messing with the sole solid. The important part of this script should look like this:

```
/prep7
et,1,45
blc4,0,0,1,1,1
esize,.1
vmesh,1
```

```
PSMESH,1,junk,,all, ,1,X, ,node(0.75,0,0)
```



Don't forget to delete the pretension elements when you're done, and also the element type and extra node made by PSMESH if you want to be tidy! Such as:

```
ESEL,S,ENAME,,179
EDELE,ALL
ETDELE,2
NDELE,1332 !This node was auto-created by PSMESH
```

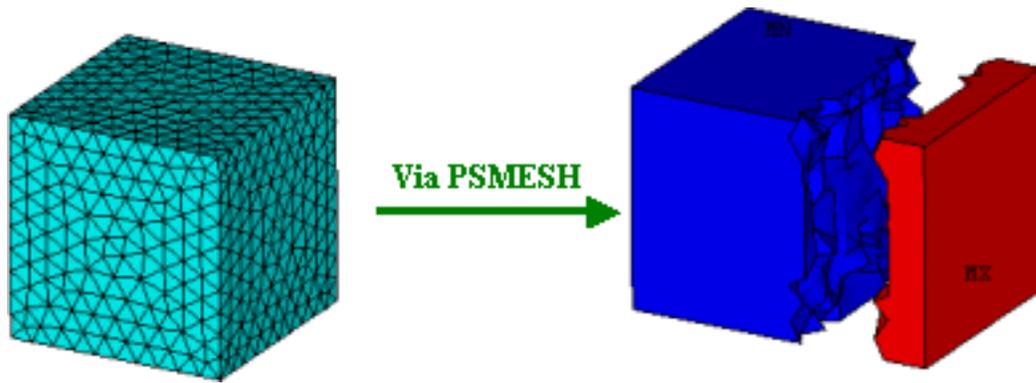
You can use this new surface normally for internal loads, contact, etc. Note that the split will be an approximate plane, but no elements will be changed. Thus, if you have a tetrahedral mesh, the split surface will be "jagged" like this:



The Focus



A Publication for ANSYS Users



So next time you need a boolean *after* meshing and applying all your BCs, you might pirate the PSMESH command.

So if I must say so myself, pretty clever, huh?



The Focus



A Publication for ANSYS Users

C Descriptors, C Descriptors Run

by Jeff Strain, PADT

For years, ANSYS users have had the ability to write data to a text file. This was done via the *VWRITE or *MWRITE command followed by a line of Fortran descriptors enclosed in parentheses. The main disadvantages to the Fortran descriptor method are that you cannot write integers (although the TL1 — tab left one space—descriptor provides a workaround to this) and you cannot output character parameters longer than eight characters (again, the multiple character parameters and space suppression provide a workaround).

With C descriptors, you can write out integers and character strings up to 128 characters with no workarounds. Plus you don't need to waste your precious time typing out parentheses and single quotes as you do with Fortran descriptors (hey, for people who type the way I do, this is significant). The basic C descriptors are as follows:

- %I Integer data
- %F Floating point format
- %G General numeric format
- %E Scientific notation
- %C, Character strings (up to 128 characters) or arrays;
- %S No difference in functionality that I can discern
- %/ Line break
- %% Single percent sign

To use these, simply place after the *VWRITE or *MWRITE line as you would with Fortran descriptors, except with no parentheses.

More than likely, you will want to use the enhanced format descriptors. These allow you to control the width of your columns and the precision of your data. Enhanced C descriptors available in ANSYS include the following:

- %wI w is the column width. Integer is preceded by the number of blank characters needed to fill the column.
- %0wI Same as above except integer is padded by zeroes instead of spaces.
- %0w.pIPad integer with zeros as necessary to obtain a minimum of p digits.
- %w.pF w is the column width. Floating point format to p decimal places.
- %w.pG General format with p significant digits.
- %w.pE Scientific notation to p decimal places.
- %wC, Character string with column width of w.
- %wS
- %-wC, Left justify character string (right justify is default).
- %-wS
- %wX Include w blank characters.

Example: Consider array NODELOCS, displayed in Figure 1.

The Focus



A Publication for ANSYS Users

Array Parameter NODELOCS		
File Edit Help		
Page Increment	Full Page View	
Initial Constant	0	
Selected: NONE		
	1	2
1	0.17335952	0.25464970
2	2.74301406	1.18992747
3	1.02991104	0.56640896
4	1.88646255	0.87816821
5	2.15721906	2.79938600
6	2.45011656	1.99465673
7	-0.41243547	1.86410824
8	1.30066755	2.48762675
9	0.44411604	2.17586749
10	-0.11953797	1.05937897

Figure 1. Node location data stored in array NODELOCS.

We wish to write out node numbers and x and y locations (listed in columns 1 and 2 of the array, respectively) to a file. We'll use several options to illustrate the behavior of the basic and enhanced C descriptors. Note that in the following examples, I've omitted the *CFOPEN/*CFCLOSE commands in the interest of saving space.

Let's start basic. The following script:

```
*VWRITE
Node X-Loc Y-Loc
*VWRITE,SEQU,nodelocs(1,1),nodelocs(1,2)
%I %G %G
```

yields

```
Node X-Loc Y-Loc
1 0.173359529 0.254649708
2 2.74301406 1.18992747
3 1.02991104 0.566408962
4 1.88646255 0.878168216
5 2.15721906 2.79938601
6 2.45011656 1.99465674
7 -0.412435471 1.86410824
8 1.30066755 2.48762675
9 0.44411604 2.1758675
10 -0.119537971 1.05937898
11 0.737013541 1.37113823
```

The Focus



A Publication for ANSYS Users

```
12 1.59356505 1.68289748
```

Kind of messy, isn't it? If all you're doing is importing the data into an Excel spreadsheet it's fine, in which case you may want to separate the data with commas. At least we don't have any pesky decimal points after the node numbers. Let's try it again but with enhanced descriptors.

```
*VWRITE
```

```
Node X-Loc Y-Loc
```

```
*VWRITE,SEQU,nodelocs(1,1),nodelocs(1,2)
```

```
%4I %10.3G %11.4G
```

```
yields
```

Node	X-Loc	Y-Loc
1	0.173	0.2546
2	2.74	1.190
3	1.03	0.5664
4	1.89	0.8782
5	2.16	2.799
6	2.45	1.995
7	-0.412	1.864
8	1.30	2.488
9	0.444	2.176
10	-0.120	1.059
11	0.737	1.371
12	1.59	1.683

That's better, isn't it? One issue I've run into with the %w.pG descriptor is that I have to set my column width to an unusually large value to prevent the column from being filled with asterisks (asterisks indicate that the column width is set too low). Plus, aesthetically, I tend to prefer a fixed number of decimal places over a fixed number of significant digits. I'll generally resort to scientific notation if significant digits are important and my data has varying orders of magnitude.

Let's try it with some other options to further demonstrate the enhanced C descriptors. Note that I would never output this data in this format if this were a real life project; I would use floating point format for both the x and y locations.

```
*VWRITE
```

```
Node X-Loc Y-Loc
```

```
*VWRITE,SEQU,nodelocs(1,1),nodelocs(1,2)
```

```
%04.2I %10.3E %7.4F
```

```
yields
```

The Focus



A Publication for ANSYS Users

Node	X-Loc	Y-Loc
01	1.734E-01	0.2546
02	2.743E+00	1.1899
03	1.030E+00	0.5664
04	1.886E+00	0.8782
05	2.157E+00	2.7994
06	2.450E+00	1.9947
07	-4.124E-01	1.8641
08	1.301E+00	2.4876
09	4.441E-01	2.1759
10	-1.195E-01	1.0594
11	7.370E-01	1.3711
12	1.594E+00	1.6829

I hope that the information in the article has opened some doors for you when it comes to exporting data. If you'd like to learn more about this and other APDL (ANSYS Parametric Design Language) topics, consider ordering our [ANSYS Customization with APDL](#) guide. This guide is a compilation of the course notes for our [APDL class](#) and is intended for the new or veteran APDL programmer.

The Focus



A Publication for ANSYS Users

Improved Meshtool GUI

by Rod Scholl, PADT

I'll keep this one short and sweet. [Deeptiranjana Barik](#) has kindly offered up his TCL-generated Meshtool for you, your co-workers, friends, pets, etc. He should have sent an anonymous request for an improved Meshtool then wowed us with this submission, subsequently taking the APDL Guru title. Instead, he will have to settle for an "honorary TCL Guru" status. We're not worthy.

A screen capture is shown to the right: Besides the expected Mesh tools, you'll note the full host of Boolean operations as well as a compact format for manipulating the working plane. Also note the KPDIS, and friends at the bottom.

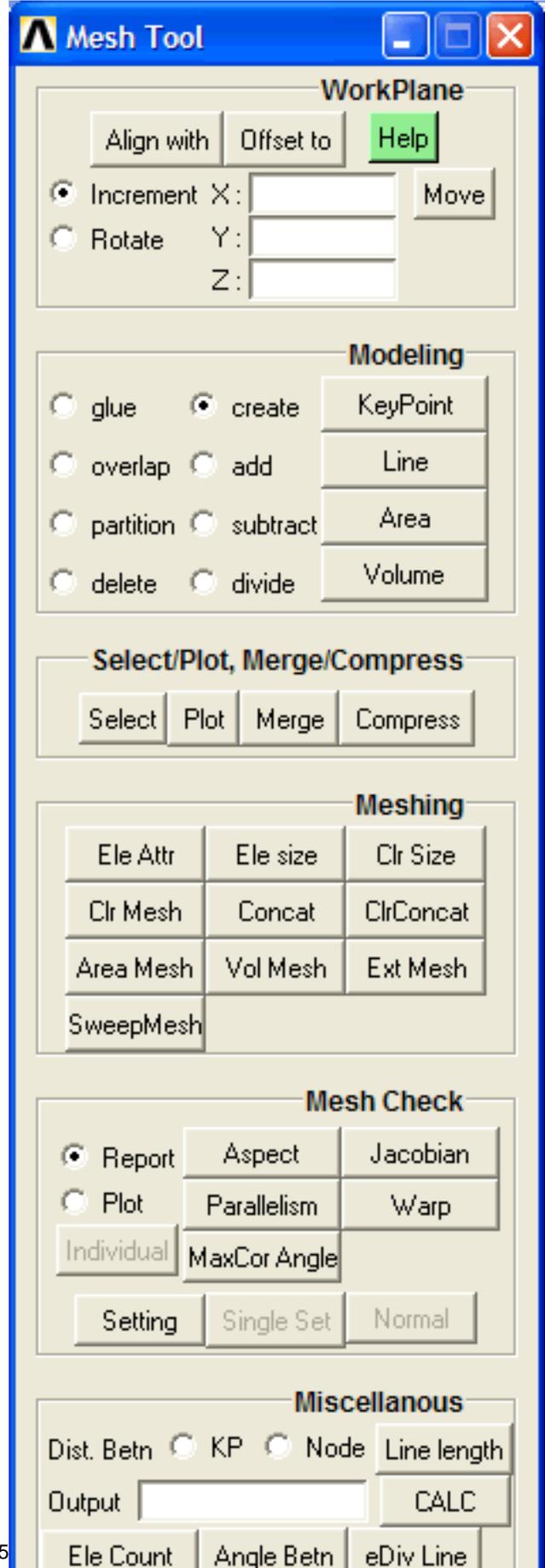
At least half my meshes are GUI generated. It's always a gamble on whether or not scripted code makes sense when meshing, and often I gamble to forgo the scripting and just get the job done quick and dirty. Well, quick anyway...

Is that a calculator, too? I always feel like a goof entering $a=32.17*12$ and then checking the output window. 386.04 or was it 384.06? Why is it the same brain cells I keep killing?

So as can see, not only is the power of TCL programming highlighted here as well as the customization options of ANSYS (learn more from Matt's article and tcl resource pages), but also the tool itself is thorough and refreshingly compact. Want to add another button — go on dive in and edit the TCL files. You know you want to. Before you know it you might program a Contact Sorcerer to banish that Contact Wizard back to depths from which it rose.

Thanks again to Deeptiranjana Barik! Sharing with the user community is often thankless even though it is appreciated.

Download [Meshtool.zip](#) (15 kB) to use this tool. Be sure to read the Readme file contained within the zip archive.



The Focus



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 '05	4/25	4/27	152	AI*Environment	Tempe, AZ
May '05	5/2	5/4	101	Introduction to ANSYS, Part I	Irvine, CA
	5/5	5/6	102	Introduction to ANSYS, Part II	Irvine, CA
	5/9	5/10	100	Engineering with Finite Element Analysis	Tempe, AZ
	5/12	5/13	803	Tcl/Tk for ANSYS	Tempe, AZ
	5/16	5/17	604	Introduction to CFX	Irvine, CA
	5/19	5/20	203	Dynamics	Irvine, CA
Jun '05	5/23	5/23	106	ANSYS Workbench DesignXplorer	Tempe, AZ
	6/13	6/15	101	Introduction to ANSYS, Part I	Tempe, AZ
	6/20	6/21	201	Basic Structural Nonlinearities	Albuquerque, NM
	6/22	6/23	204	Advanced Contact and Bolt Pretension	Albuquerque, NM
	6/27	6/28	301	Heat Transfer	Irvine, CA
	6/29	6/30	107	ANSYS Workbench DesignModeler	Irvine, CA
Jul '05	7/11	7/13	104	ANSYS Workbench Simulation - Intro	Albuquerque, NM
	7/14	7/15	801	ANSYS Customization with APDL	Albuquerque, NM

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.



The Focus



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.