Operations Guide

ANSYS Release 8.1

001972
April 2004

ANSYS, Inc. is a
UL registered
ISO 9001: 2000
Company

Operations Guide

ANSYS Release 8.1

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
001619	ANSYS 6.1	April 2002
001695*	ANSYS 7.0	October 2002
001788*	ANSYS 7.1	May 2003
001901*	ANSYS 8.0	October 2003
001972*	ANSYS 8.1	April 2004

* ANSYS Documentation on CD.

Trademark Information

Table of Contents

List of Figures

List of Tables

Chapter 1: Introducing ANSYS

1.1. What ANSYS Is

ANSYS finite element analysis software enables engineers to perform the following tasks:

- Build computer models or transfer CAD models of structures, products, components, or systems.
- Apply operating loads or other design performance conditions.
- Study physical responses, such as stress levels, temperature distributions, or electromagnetic fields.
- Optimize a design early in the development process to reduce production costs.
- Do prototype testing in environments where it otherwise would be undesirable or impossible (for example, biomedical applications).

The ANSYS program has a comprehensive graphical user interface (GUI) that gives users easy, interactive access to program functions, commands, documentation, and reference material. An intuitive menu system helps users navigate through the ANSYS program. Users can input data using a mouse, a keyboard, or a combination of both.

1.2. Purpose of This Manual

This manual provides basic instructions for operating the ANSYS program: starting and stopping the product, using and customizing its GUI, using the online help system, etc. For other information about using ANSYS, see the following documents:

- For general instructions on performing finite element analyses for any engineering discipline, see the *ANSYS Basic Analysis Guide*, the *ANSYS Modeling and Meshing Guide*, and the *ANSYS Advanced Analysis Techniques Guide*.
- For information about performing specific types of analysis (thermal, structural, etc.), see the applicable *Analysis Guide*.
- For examples of analyses, see the *ANSYS Tutorials* and *ANSYS Verification Manual*.
- For reference information about ANSYS commands, elements, and theory, see the *ANSYS Commands Reference*, *ANSYS Elements Reference*, and *ANSYS, Inc. Theory Reference*.

Chapter 2: The ANSYS Environment

2.1. Organization of the ANSYS Program

The ANSYS program is organized into two basic levels:

- Begin level
- Processor (or Routine) level

The *Begin level* acts as a gateway into and out of the ANSYS program. It is also used for certain global program controls such as changing the jobname, clearing (zeroing out) the database, and copying binary files. When you first enter the program, you are at the Begin level.

At the *Processor level*, several processors are available. Each processor is a set of functions that perform a specific analysis task. For example, the general preprocessor (PREP7) is where you build the model, the solution processor (SOLUTION) is where you apply loads and obtain the solution, and the general postprocessor (POST1) is where you evaluate the results of a solution. An additional postprocessor, POST26, enables you to evaluate solution results at specific points in the model as a function of time.

2.2. Entering a Processor

In general, you enter a processor by selecting it from the ANSYS Main Menu in the Graphical User Interface (GUI). For example, choosing **Main Menu> Preprocessor** takes you into PREP7. Alternatively, you can use a command to enter a processor (the format is `/name`, where *name* is the name of the processor). Table 2.1: "Processors (Routines) Available in ANSYS" lists each processor, its function, and the command to enter it.

Table 2.1 Processors (Routines) Available in ANSYS

Processor	Function	GUI Path	Command
PREP7	Build the model (geometry, materials, etc.)	**Main Menu> Preprocessor**	**/PREP7**
SOLUTION	Apply loads and obtain the finite element solution	**Main Menu> Solution**	**/SOLU**
POST1	Review results over the entire model at specific time points	**Main Menu> General Postproc**	**/POST1**
POST26	Review results at specific points in the model as a function of time	**Main Menu> TimeHist Postpro**	**/POST26**
OPT	Improve an initial design	**Main Menu> Design Opt**	**/OPT**
PDS	Quantify the effect of scatter and uncertainties associated with input variables of a finite element analysis on the results of the analysis	**Main Menu> Prob Design**	**/PDS**
AUX2	Dump binary files in readable form	**Utility Menu> File> List> Binary Files** **Utility Menu> List> Files> Binary Files**	**/AUX2**
AUX12	Calculate radiation view factors and generate a radiation matrix for a thermal analysis	**Main Menu> Radiation Matrix**	**/AUX12**

Processor	Function	GUI Path	Command
AUX15	Translate files from a CAD or FEA program	**Utility Menu> File> Import**	**/AUX15**
RUNSTAT	Predict CPU time, wavefront requirements, etc. for an analysis	**Main Menu> Run-Time Stats**	**/RUNST**

2.3. Exiting from a Processor or ANSYS

To return to the Begin level from a processor, pick **Main Menu> Finish** or issue the **FINISH** (or **/QUIT**) command. You can move from one processor to another without returning to the Begin level. Simply pick the processor you want to enter, or issue the appropriate command.

To leave the ANSYS program (and return to the system level), pick **Utility Menu> File> Exit**, to display the **Exit from ANSYS** dialog box, or use the **/EXIT** command. By default, the program saves the model and loads portions of the database automatically and writes them to the database file, **Jobname.DB**. If a backup of the current database file already exists, ANSYS writes it to **Jobname.DBB**. Options in the dialog box (and on the **/EXIT** command) allow you to save other portions of the database or to quit without saving.

2.3.1. Stopping the Input of a File

You can also stop the processing of an ANSYS file as it is being input. Most files of more than a few lines will display the **ANSYS Process Status** window at the top of the screen. If you want to terminate the input of a file, select the **STOP** button on the **ANSYS Process Status** window. ANSYS itself does not stop when you select the **STOP** button. Stopping file input is useful if you inadvertently input a binary file.

To input a new file, select **Utility Menu> File> Clear & Start New** to clear the current file from memory, then select a file to input. If you want to return to processing the original file, select **Utility Menu> File> Read Input from...** and select the name of the file, the line number or label to resume from, and select the **OK** button. See the **/INPUT** command for more information on resuming a file input process.

2.4. The ANSYS Database

In one large database, the ANSYS program stores all input data (model dimensions, material properties, load data, etc.) and results data (displacements, stresses, temperatures, etc.) in an organized fashion. The main advantage of the database is that you can list, display, modify, or delete any specific data item quickly and easily.

No matter which processor you are in, you are working with the same database. This gives you basic access to the model and loads portions of the database *from anywhere in the program*. "Basic access" means the ability to select, list, or display an item.

2.5. Defining or Deleting Database Items

To define items, or to delete items from the database, you must be in the appropriate processor. For example, you can define nodes, elements, and other geometry only in PREP7, the general preprocessor. You can specify and apply loads in either the PREP7 or the SOLUTION processor, and you can declare optimization variables only in OPT (the design optimization processor). However, you can select geometry items, list them, or display them from anywhere in the program, including the Begin level.

2.6. Saving the Database

Because the database contains all your input data, you should frequently save copies of it to a file. To do this, pick **Utility Menu> File> Save as Jobname.DB** or issue the **SAVE** command. Either choice writes the database to the file **Jobname.DB**. If you use the **SAVE** command, you have the option to save:

- the model data only
- the model and solution data
- the model, solution and preprocessing data

To specify a different file name, pick **Utility Menu> File> Save as** or use the appropriate fields on the **SAVE** command. Any save operation first writes a backup of the current database file (if the database already exists) to **Jobname.DBB**. If a **Jobname.DBB** file already exists, the new backup file overwrites it. For a static or transient structural analysis, the file **Jobname.RDB** (a copy of the database) will be automatically saved at the first substep of the first load step.

2.7. Restoring Database Contents

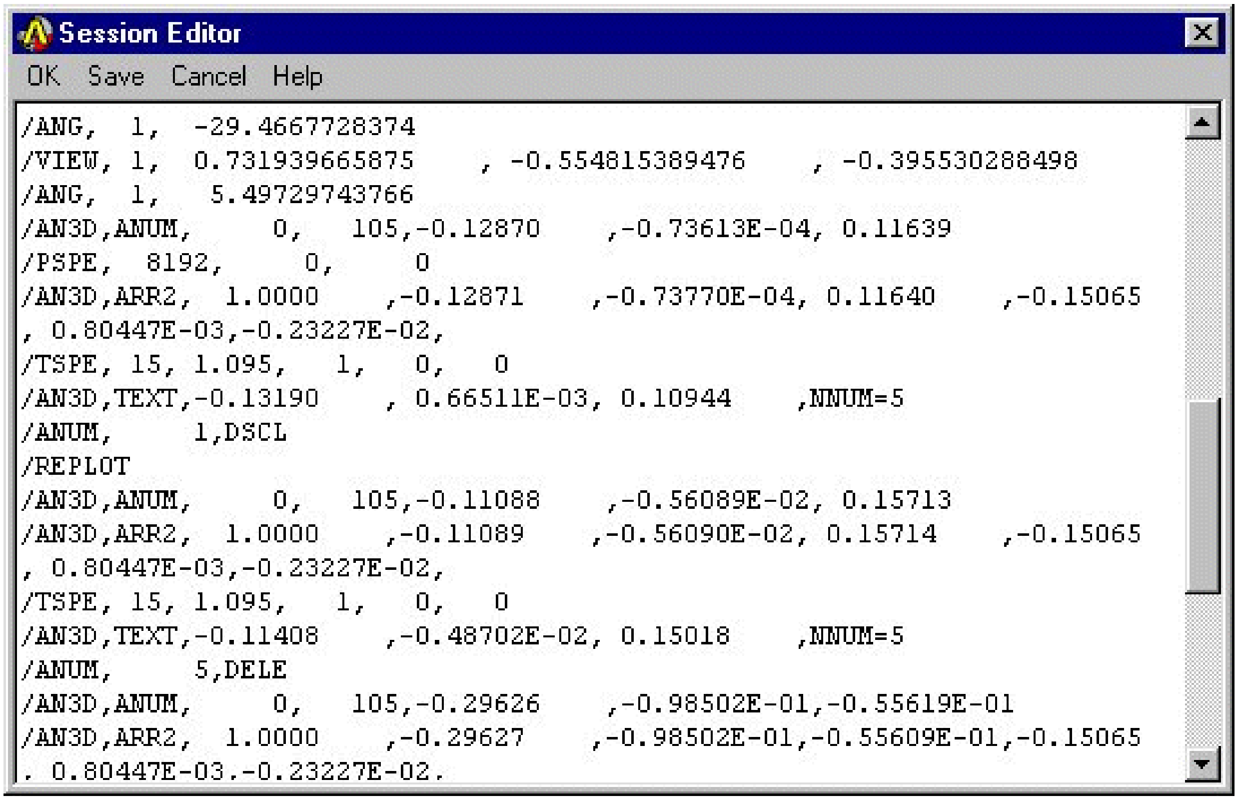
To *restore data* from the database file, pick **Utility Menu> File> Resume Jobname.DB** or issue the **RESUME** command. This reads the file **Jobname.DB**. To specify a different file name, pick **Utility Menu> File> Resume from** or use the appropriate fields on the **RESUME** command.

You can save or resume the database from anywhere in the ANSYS program, including the Begin level. A resume operation replaces the data that currently is in memory with the data in the named database file. Using the save and resume operations together is useful when you want to "test" a function or command. When you do a multiframe restart, **ANTYPE**,,REST automatically resumes the **.RDB** file for the current job.

2.8. Using the Session Editor

During an analysis, you may want to modify or delete commands entered since your last **SAVE** or **RESUME**. You can access the session editor by issuing the **UNDO** command, or by choosing **Main Menu> Session Editor**. The session editor display is shown below.

Figure 2.1 The Session Editor



Use this dialog for displaying and editing the string of operations performed since your last **SAVE** or **RESUME** command. You can modify command parameters, delete whole sections of text and even save a portion of the command string to a separate file.

You can access the following file operations from the session editor dialog:

- **OK**: Enters the series of operations displayed in the window below. You will use this option to input the command string after you have modified it.
- **Save**: Saves the command string displayed in the window below to a separate file. ANSYS names the file **Jobnam000.cmds**, with each subsequent save operation incrementing the filename by one digit. You can use the **/INPUT** command to reenter the saved file.
- **Cancel**: Dismisses this window and returns to your analysis.
- **Help**: Displays the command reference for the **UNDO** command.

The Session Editor is available in interactive (GUI) mode only. If no **SAVE** or **RESUME** command has been issued during your analysis, all commands from your current session will be executed, including your **start81.ans** file, if present.

2.9. Clearing the Database

While building a model, sometimes you may want to clear out the database contents and start over. To do so, choose **Utility Menu> File> Clear & Start New** or issue the **/CLEAR** command. Either method clears (zeros out) the database stored in memory. Clearing the database has the same effect as leaving and reentering the ANSYS program, but does not require you to exit.

2.10. ANSYS Program Files

The ANSYS program writes and reads many files for data storage and retrieval. File names follow this pattern:

```
Name.Ext
```

Name defaults to the jobname, which you can specify while entering the ANSYS program or by choosing **Utility Menu> File> Change Jobname** (equivalent to issuing the **/FILNAME** command). The default jobname is **FILE** (or **file**).

Ext is a unique, two- to four-character ANSYS identifier that identifies the contents of the file. For example, **Jobname.DB** is the database file, **Jobname.EMAT** is the element matrix file, and **Jobname.GRPH** is the neutral graphics file. Some systems (such as PCs) truncate the extension to three characters. Also, the extension may be in lowercase, depending on the system.

2.11. ANSYS File Types

Table 2.2: "ANSYS File Types and Formats" lists the main ANSYS file types and their formats. For more information about files, see Chapter 18, "File Management and Files" in the *ANSYS Basic Analysis Guide*.

Table 2.2 ANSYS File Types and Formats

File Type	File Name	File Format
Log file	**Jobname.LOG**	ASCII
Error file	**Jobname.ERR**	ASCII
Output file	**Jobname.OUT**	ASCII
Database file	**Jobname.DB**	Binary
Results file: structural or coupled thermal magnetic FLOTRAN	**Jobname.xxx** **Jobname.RST** **Jobname.RTH** **Jobname.RMG** **Jobname.RFL**	Binary
Load step file	**Jobname.Sn**	ASCII
Graphics file	**Jobname.GRPH**	ASCII (special format)
Element matrices	**Jobname.EMAT**	Binary

On the following ANSYS commands, you can specify the name and path of the file to be written:

/ASSIGN	***LIST**
/COPY	**/OUTPUT**
***CREATE**	**/PSEARCH**
/DELETE	**/RENAME**
/INPUT	

In such cases, the filename can contain up to 248 characters, including the directory name, and the extension can contain up to eight characters. If the file name uses more than 248 characters, including the directory, you must use a soft link on Unix systems.

ANSYS can process blanks in file or directory names, so blank spaces are allowed in ANSYS object names. Be aware that many UNIX commands do not support object names with spaces. When an object has a blank space in its name, always enclose the name in a pair of single quotes.

On UNIX systems, all directory names except for /(root) should end with a slash (/). For example, to run the ANSYS program using an input file called **vm1.dat**, which resides in the directory **/ansys_inc/v81/ansys/data/verif**, use the following commands:

```
ansys81
/inp,vm1,dat, /ansys_inc/v81/ansys/data/verif/
```

On Windows systems, you must use back slashes (\) instead of a slash in directory names. For example, on a Windows system, the directory path shown in the UNIX example above looks like this:

```
/inp,vm1,dat, Program Files\Ansys Inc\V81\ANSYS\data\verif\
```

2.12. ANSYS File Sizes

The maximum size of an ANSYS file depends on the system limit and on the ability of ANSYS to handle large files on that system. Most computer systems now handle very large files without any need for the automatic file splitting option that is provided in ANSYS. Linux 32-bit systems and earlier Windows systems (FAT and FAT32) still have some limitations that require file splitting in some instances. ANSYS default settings will handle such cases. The sparse solver files (**jobname.LN*xx***) are automatically split at just under 2 GB on all 32-bit operating systems (Linux and Windows) and at just under 8 GB on all other systems. (You cannot increase the size of the split file partitions for sparse solver files. This limitation applies only to files used by the sparse solver.)

2.13. The Jobname.LOG File

The **Jobname.LOG** file (also called the session log) is especially important, because it provides a complete log of your ANSYS session. The file opens immediately when you enter the ANSYS program, and it records all commands you execute, whether you execute those commands via GUI paths or type them in directly. You can read the **Jobname.LOG** file, view it while in ANSYS, edit it, and input it later.

The ANSYS program always appends log data to the log file instead of overwriting it. If you change the jobname while in an ANSYS session, the log file name *does not* change to the new jobname. For more information about **Jobname.LOG**, see Chapter 8, "Using the ANSYS Command Log".

2.14. Communicating With the ANSYS Program

The easiest way to communicate with the ANSYS program is by using the ANSYS menu system, called the Graphical User Interface (GUI).

2.14.1. Communicating Via the Graphical User Interface (GUI)

The GUI consists of windows, menus, dialog boxes, and other components that allow you to enter input data and execute ANSYS functions simply by picking buttons with a mouse or typing in responses to prompts. All users, both beginner and advanced, should use the GUI for interactive ANSYS work. See Chapter 4, "Using the ANSYS GUI " for an extensive discussion of how to use the GUI. The rest of this section describes other topics related to communication with ANSYS commands, abbreviations, etc.

2.14.2. Communicating Via Commands

Commands are the instructions that direct the ANSYS program. ANSYS has more than 1200 commands, each designed for a specific function. Most commands are associated with specific (one or more) processors, and work only with that processor or those processors.

To use a function, you can either type in the appropriate command or access that function from the GUI (which internally issues the appropriate command). The *ANSYS Commands Reference* describes all ANSYS commands in detail, and also tells you whether each command has an equivalent GUI path. (A few commands do not.)

ANSYS commands have a specific format. A typical command consists of a command name in the first field, usually followed by a comma and several more fields (containing arguments). A comma separates each field. For example, the **F** command, which applies a force at a node, looks like this:

```
F,NODE,Lab,VALUE
```

To apply an X-direction force of 2000 at node number 376, the **F** command would read as follows:

```
F,376,FX,2000
```

You can abbreviate command names to their first four characters (except as noted in the *ANSYS Commands Reference*). For example, **FINISH**, **FINIS**, and **FINI** all have the same meaning. Some "commands" (such as **ADAPT** and **RACE**) are actually *macros*. You must enter macro names in their entirety.

> *Note* — If you are not sure whether an instruction is a command or a macro, see the *ANSYS Commands Reference*.

Commands that begin with a *slash* (**/**) usually perform general program control tasks, such as entry to routines, file management, and graphics controls. Commands that begin with a *star* (*****) are part of the ANSYS Parametric Design Language (APDL). See the *ANSYS APDL Programmer's Guide* for details.

Command *arguments* may take a number or an alphanumeric label, depending on their purpose. In the **F** command example described previously, *NODE* and *VALUE* are numeric arguments, but *Lab* is an alphanumeric argument. In this and other ANSYS manuals, numeric arguments appear in all uppercase italic letters (as in *NODE* and *VALUE*), and alphanumeric arguments appear in initial uppercase italic format (as in *Lab*). Some commands (for example, **/PREP7**, **/POST1**, **FINISH**, etc.) have no arguments, so the entire command consists of just the command name.

Some general rules and guidelines for commands are listed below:

- When you enter commands, the arguments do not have to be in specific columns.
- You can use successive commas to skip arguments. When you do so, ANSYS uses default values for the omitted arguments (as discussed in the individual command descriptions).
- You can string together multiple commands on the same line by using the $ character as the delimiter for each command. (For restrictions on use of the $ delimiter, see the *ANSYS Commands Reference*.)
- The maximum number of characters allowed per line is 640, including commas, blank spaces, $ delimiters, and any other special characters.

  > *Note* — Other software programs and printers may wrap text to the next line or truncate the text after a certain character.

- Real number values input to integer data fields will be rounded to the nearest integer. The absolute value of integer data must fall between zero and 2,000,000,000.

- The acceptable range of values for real data is +/-1.0E+60 to +/-1.0E-60. No exponent can exceed +60 or be less than -60. The program accepts real numbers in integer fields, but rounds them to the nearest integer. You can specify a real number using a decimal point (such as 327.58) or an exponent (such as 3.2758E2). The E (or D) character, used to indicate an exponent, may be in upper or lower case. This limit applies to all ANSYS input commands, regardless of platform.

Even though all ANSYS input must be within the allowed range, all numeric operations, including parametric operations, can produce numbers to machine precision, which may exceed the ANSYS input range.

- ANSYS interprets numbers entered for *Angle* arguments as degrees. Note that there are functions in ANSYS that *could* use radians if the ***AFUN** command had been used.
- The following special characters are not allowed in alphanumeric arguments:

```
! @ # $ % ^ & * ( ) _ - +
= | \ { } [ ] " ' / < > ~ `
```

  Exceptions are filename and directory arguments, where some of these characters may be required to specify system-dependent pathnames. However, using special characters in filename and directory arguments could result in ANSYS or the operating system misreading the argument. We strongly recommend that you limit filename and directory arguments to A-Z, a-z, 0-9, -, _, and spaces. Any text prefaced by an exclamation mark (!) is treated as a comment.
- Avoid using tabs (to line up comments, for instance) or other control (CTRL) sequences. They usually generate device-dependent characters that the program cannot recognize.
- If you are a long time ANSYS user, avoid using commands that have been removed from the currently documented command set. Such commands are obsolete, and may cause difficulties.

2.14.3. Command Defaults

To minimize the amount of data input, most commands have defaults. There are two types of defaults: command default and argument default.

A *command default* is the specification assumed when a command is not issued. For example, if you do not issue the **/FILNAME** command, the jobname defaults to **FILE** (or whatever jobname was specified when you entered the ANSYS program).

An *argument default* is the value assumed for a command argument if the argument is not specified. For example, if you issue the command **N**,10 (defining node 10 with the *X, Y, Z* coordinate arguments left blank), the node is defined at the origin; that is, *X, Y*, and *Z* default to zero. Numeric arguments (such as *X, Y, Z*) default to zero except as noted in the *ANSYS Commands Reference*. The command descriptions usually explain defaults for other arguments.

> *Note* — The defaults for some commands and their arguments differ depending on which ANSYS product is using the commands. The "Product Restrictions" section of the descriptions of the affected commands clearly documents such cases. If you plan to use your input file in more than one ANSYS product, you should explicitly specify commands or command argument values, rather than letting them default. Otherwise, behavior in the other ANSYS product may be different from what you expect.

2.14.4. Abbreviations

If you use a command or a GUI function frequently, you can rename it or abbreviate it to a string of up to eight alphanumeric characters using one of the following:

Command(s): *ABBR
GUI: Utility Menu> Macro> Edit Abbreviations

Utility Menu> MenuCtrls> Edit Toolbar

For example, the following command defines ISO as an abbreviation for the command **/VIEW**,,1,1,1 (which specifies isometric view for subsequent graphics displays):

```
*ABBR,ISO,/VIEW,,1,1,1
```

Keep the following rules and guidelines in mind when creating abbreviations:

- The abbreviation must begin with a letter and should not have any spaces.
- If an abbreviation that you set matches an ANSYS command, the abbreviation overrides the command. Therefore, *use caution in choosing abbreviation names*.
- You can abbreviate up to 60 characters, and up to 100 abbreviations are allowed per ANSYS session.

In the GUI, abbreviations appear as push buttons on the Toolbar, which you can execute with a quick pick of the mouse. For details, see the section on using the toolbar in Chapter 4, “Using the ANSYS GUI ”.

2.14.5. Command Macro Files

You can record a frequently used sequence of ANSYS commands in a macro file, thus creating a personalized ANSYS command. If you enter a command name that ANSYS does not recognize, it searches for a macro file by that name (with an extension of **.MAC** or **.mac**). If the file exists, ANSYS executes it.

On Unix and Windows systems, the ANSYS program searches for macro files in the following order:

- ANSYS looks first in the ANSYS APDL directory.
- It then looks at the directories that have been defined for the environmental variable **ANSYS_MACROLIB**. You can set up the **ANSYS_MACROLIB** variable after the installation of ANSYS software and before the program is started.

  On Unix, the structure for **ANSYS_MACROLIB** is:

  ```
  dir1/:dir2/:dir3/
  ```

  On Windows, the structure is:

  ```
  c:\dir1\;d:\dir2\;e:\dir3
  ```

  The letter to the left of the colon indicates the drive where the directory is stored.

  Enter up to 255 characters for the entire string. *`Dir1`* is searched first, followed by *`dir2`*, *`dir3`*, etc. These files provide customization at both the site and user levels.
- Next, on Unix systems, ANSYS looks in **/PSEARCH** or in the login directory. On Windows systems, it looks in **/PSEARCH** or in the home directory.
- Finally, ANSYS looks in the current or working directory.

ANSYS searches for both upper and lower case macro file names in each search directory, except **/apdl** on Unix systems. If both exist in the search directory, the upper case file is used. Only upper case is used in the **/apdl** directory on Unix systems.

The ANSYS installation media provide many ANSYS macro files that reside in the **/apdl** subdirectory. If you cannot use any of the ANSYS-provided macro files, contact your system administrator.

To access any macro, you simply enter its file name. For instance, to access the **LSSOLVE.MAC** file, you enter **LSSOLVE**. You can also access macros you created via the **Utility Menu> Macro> Execute Macro** menu path. However, this menu path will not work for any macros containing function granules (such as a call to a dialog box) or picking commands. Macros with these functions must be accessed by entering the macro name in the Input Window.

Specifying File Names in Windows

In the Windows environment, some devices/ports have specific names, such as PRN, COM1, COM2, LPT1, LPT2, and CON. The device/port names resemble files in that they can be opened, read from, written to, and closed. Entering the names of these devices/ports in ANSYS, however, causes unpredictable behavior, including system freezes or fatal error conditions. Therefore, do not issue PC device/port names as commands.

Configuring Search Paths on Windows Systems

1. In the Control Panel, click on the System Icon.
2. On Windows XP systems, click on My Computer on the Start Menu. Under **System Tasks**, select **View System Information**. Select the Advanced Tab. Click on the **Environment Variables** button. Click **New** under System Variable. Enter the value of **ANSYS_MACROLIB** for the variable name. Enter

   ```
   <drive>:\<dir>\;<drive>:\<dir2>\;<drive>:\<dir3>\;
   ```

   for the variable value. Click **OK**.
3. On Windows 2000 systems, select the Advanced tab. Click on the **Environment Variables** button. Click on the **New** button under **System Variables**. Enter the value of ANSYS_MACROLIB for the variable name. Enter

   ```
   <drive>:\<dir>\;<drive>:\<dir2>\;<drive>:\<dir3>\;
   ```

   for the variable value. Click on the **OK** button.

Chapter 3: Running the ANSYS Program

You can run the ANSYS program in interactive mode or in batch mode. In interactive mode (the default), you exchange information with the computer continuously. You can execute a command by selecting its menu path in the GUI or by typing it in directly. The ANSYS program processes the command in real time. Interactive mode allows you to use the GUI, online help, and various tools to create the engineering model in the graphics window and modify it as you work through the analysis.

In batch mode, you submit a file of commands to the ANSYS program. This command file may have been generated by a previous ANSYS session, by another program, or by creating a command file with an editor. On some operating systems, you can run a batch job in the background while doing other work on the computer. Batch mode is useful when you do not need to interact with the program, such as during the solution phase of an analysis.

ANSYS offers a set of separately-licensed "batch-only" products. If you run a batch-only product, you do not have access to the GUI, to online help, or to other interactive features of ANSYS. For more information about the batch-only products, see your ANSYS sales representative.

There are a number of options for starting and setting options for your ANSYS session. You can enter the ANSYS program directly by issuing the ANSYS execution command at the command level and define aspects of the ANSYS operating system using the optional arguments for the execution command.

You can also use the ANSYS Launcher to set options for an ANSYS run and its auxiliary programs, such as the LS-DYNA solver.

3.1. Starting an ANSYS Session from the Command Level

To start an interactive ANSYS session in graphics mode on UNIX systems, type the following command at the prompt:

```
ansys81 -g
```

To start ANSYS from the MS-DOS command prompt on a Windows system, type the following command:

```
ansys81
```

Always use the ANSYS-supplied scripts (**ansys81** and **launcher81**) for running ANSYS. User-written scripts for running ANSYS products are not supported.

You can specify any of the following options for the ANSYS execution command:

`-ansexe`	In the ANSYS Workbench environment, activates a custom ANSYS executable.
`-b` *`list`* or *`nolist`*	Activates the ANSYS program in batch mode. The options `-b`*`list`* or `-b` by itself cause the input listing to be included in the output. The `-b`*`nolist`* option causes the input listing not to be included. For more information about running ANSYS in batch mode, see Section 3.4: Batch Mode.
`-d` *`device`*	Specifies the type of graphics device. This option applies only to interactive mode. For UNIX systems, graphics device choices are X11, X11C, or 3D. For Windows systems, graphics device options are WIN32 or WIN32C, or 3D.
`-db` *`value`*	Defines the portion of workspace (memory) to be used for the database. The default is 256 MB.
`-dir`	Defines the initial working directory. Using the `-dir` option overrides the **ANSYS81_WORKING_DIRECTORY** environment variable.

`-dtm`	Enables the Drop Test Module (DTM) advanced task (add-on). The DTM is an optional add-on feature to the ANSYS LS-DYNA product which simplifies the procedure for simulating a drop test. This option is only valid if you have a license for the DTM. See Chapter 17, "Drop Test Module" in the *ANSYS LS-DYNA User's Guide* for more information.
`-f` *option*	Sets ANSYS to run in fixed-memory mode. The `nogrow` option sets ANSYS to use a fixed-mode memory addressing scheme. The `no` or `off` options set ANSYS to use dynamic memory allocation for scratch memory as needed (default setting). If you do not set any arguments, ANSYS uses fixed-mode memory allocation for scratch memory, but allows other sections of the ANSYS work space to grow as required.
`-fs`	Enables the ANSYS Frequency Sweep VT advanced task (add-on).
`-fxs`	Enables ANSYS DesignXplorer VT advanced task (add-on).
`-g`	Launches the ANSYS program with the Graphical User Interface (GUI) on. If you select this option, an X11 graphics device is assumed for UNIX unless the `-d` option specifies a different device. This option is not used on Windows systems. To activate the GUI once ANSYS has started, you need to enter two commands in the ANSYS input window: **/SHOW** to define the graphics device, and **/MENU**,ON to activate the GUI. The `-g` option is valid only for interactive mode. *Note* — If you start ANSYS via the `-g` option, the program ignores any **/SHOW** command in the **start81.ans** file and displays a splash screen briefly before opening the GUI windows.
`-i`*inputname*	Specifies the name of the file to read input into ANSYS for batch processing.
`-j` **Jobname**	Specifies the initial jobname, a name assigned to all files generated by the program for a specific model. If you omit the `-j` option, the jobname is assumed to be **file**.
`-l` *language*	Specifies a language file to use other than US English. This option is valid only if you have a translated message file in an appropriately named subdirectory in **/ansys_inc/v81/ansys/docu** (or **Program Files\Ansys Inc\V81\ANSYS\docu** on Windows systems).
`-m` *workspace*	Specifies the total size of the workspace (memory) in megabytes. If you omit the `-m` option, the default is 512 MB.
`-mpi`	Specifies the type of MPI to use. Only valid with the -pp option. See Chapter 13, "Improving ANSYS Performance and Parallel Performance for ANSYS" in the *ANSYS Advanced Analysis Techniques Guide* for more information.
-name value	Defines ANSYS parameters at program start-up. The parameter name must be at least two characters long. For details about parameters, see the *ANSYS APDL Programmer's Guide*.
`-o`*outputname*	Specifies the name of the file to store the output from a batch execution of ANSYS.
`-p` *productname*	Defines which ANSYS product will run during the session (ANSYS Multiphysics, ANSYS Structural, etc.). For more detailed information about the `-p` option, see Section 3.5: Choosing an ANSYS Product.

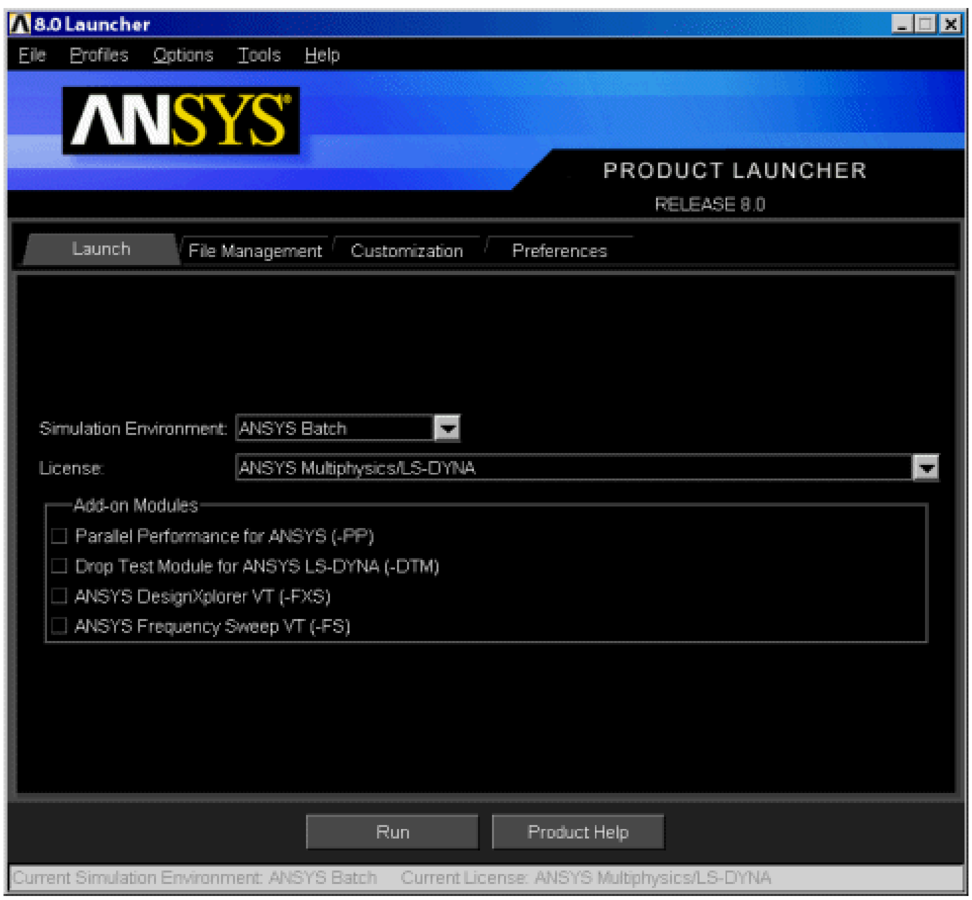
`-pp`	Enables the Parallel Performance for ANSYS advanced task (add-on). When Parallel Performance for ANSYS is licensed, you can specify two multiprocessor solver options from the **EQSLV** command: `AMG` and `DDS`. `AMG` initiates the Algebraic Multigrid solver (AMG) and `DDS` initiates the distributed solvers (DDS, DPCG, DJCG). The `-pp` option is valid only if you have a license for Parallel Performance for ANSYS. See Chapter 3, "Solution" in the *ANSYS Basic Analysis Guide* for more information about solver selection. See Chapter 13, "Improving ANSYS Performance and Parallel Performance for ANSYS" in the *ANSYS Advanced Analysis Techniques Guide* for more information about these Parallel Performance for ANSYS solvers.
`-s read` or `noread`	Specifies whether the program reads the **start81.ans** file at start-up. If you omit the `-s` option, ANSYS reads the **start81.ans** file in interactive mode and not in batch mode.
`-v`	Returns the ANSYS release number, update number, copyright date, customer number, and license manager version number.

3.2. The ANSYS Launcher

3.2.1. Starting an ANSYS Session from the Start Menu/Launcher

Use the ANSYS launcher when you want to run an ANSYS program, one of the auxiliary programs, or to access a modifiable ANSYS file. You can access some launcher functionality directly from the Windows Start Menu. To access ANSYS functionality, including the launcher, on Windows systems, choose **Start> Programs> ANSYS 8.1** and select the appropriate option.

Figure 3.1 ANSYS Launcher



To activate the launcher on UNIX systems, enter the following command:

```
launcher81
```

You can also place one of the following commands in your **.login** file or **.profile** file:

.login file command:

```
launcher81 >& /dev/null &
```

.profile file command:

```
launcher81 > /dev/null 2> &1 &
```

Either command causes the launcher to appear automatically when you log onto the system.

Start Menu Options on Windows

To activate the launcher on Windows, choose **Start> Programs> ANSYS 8.1> Configure ANSYS Products**

From the ANSYS Start Menu, you can select other options in addition to the launcher:

- Animate utility
- ANS_ADMIN utility
- Display utility
- Help

You can also run the products directly by choosing ANSYS, ANSYS Workbench, etc. from the Start Menu. If you have not yet run the launcher, selected any products, or defined any profiles, the highest product that your site is licensed for will be run. If you have run the launcher, then running the product directly from the Start Menu will bring up the product based on the last launcher configuration.

The Launcher Log File

The launcher creates a log file named **launcher.81.log**. This file contains a history of launcher selections along with any error messages that may occur. On UNIX systems, this file is written to your home directory. On Windows, it is written to the directory specified with the **TEMP** environment variable. If **TEMP** is not set, the file is written to the root directory of the value specified on the **WINDIR** environment variable. The log file is always written by default. You can turn off the log file via the **Options** menu, but we do not recommend doing so.

Launcher Tasks

Use the launcher to select the simulation environment, the specific license, and any add-on modules or analysis type you want to run. Based on your product selections, you can then specify file management, customization, preference, and LSF options. The options under each tab are explained below. You may not see all options, depending on your product selection.

The launcher tabs are:

- **Launch**
- **File Management**
- **Customization**
- **Preferences**

- **LSF**

You can also access launcher-specific functionality via the menu bar. The **File**, **Profiles**, **Options**, **Tools**, and **Help** menus are explained in later sections.

In addition to the tabs and the menu bar options, the launcher also has additional buttons at the bottom.

- **Run** launches the ANSYS product you have selected with the settings specified on the various tabs.
- **Queue** allows you to launch a batch run at a later time. When you click the Queue button, you will be prompted to enter a delayed start time and date. This option is only valid on UNIX systems.
- **LSF Run** runs your batch job via the LSF program. This option is only available if you have the LSF/Batch product installed.
- **Product Help** launches the help file for the selected simulation environment.

3.2.1.1. The Launch Tab

Here you specify your simulation environment, license, and add-on modules. The simulation environment indicates which interface you want to use to run interactively or allows you to start a batch run. Options include:

- ANSYS Workbench
- ANSYS
- ANSYS Batch
- LS-DYNA Solver

Depending on which environment you select and what licenses are available at your site, the remaining choices on the **Launch** tab as well as the options under the remaining tabs will vary.

In the **License** field, select a license from the available types. Only those licenses that are both available on your site and valid with the simulation environment selected will be shown.

You can then select from available add-on modules or, for LS-DYNA, analysis types. You will only be able to pick modules that are available for the selected environment. If a choice is grayed out, then that choice is available for the selected environment, but your site does not have the necessary license. To obtain a license for any grayed-out option, please contact your ANSYS sales representative.

Possible **Add-on Modules** include:

Parallel Performance for ANSYS
: Enables the Parallel Performance for ANSYS advanced task (add-on) to run the AMG or distributed solvers.

ANSYS Drop Test Module
: Enables the Drop Test Module (DTM) advanced task (add-on). The DTM is an optional add-on feature to the ANSYS LS-DYNA product which simplifies the procedure for simulating a drop test.

ANSYS DesignXplorer VT
: Enables the ANSYS DesignXplorer VT advanced task. The DesignXplorer VT module is part of the ANSYS Workbench Products and is used to compute the requested results for the entire range of all design parameters with one solve. See the DesignXplorer VT online help for more information.

ANSYS Frequency Sweep VT
: Enables the ANSYS Frequency Sweep VT advanced task, which provides response curves of electromagnetic applications as a function of the excitation frequency.

Possible **Analysis Types** available with the LS-DYNA Solver environment include:

- Typical LS-DYNA Analysis
- Implicit-to-Explicit Sequential Solution
- Simple Restart Analysis
- Small Restart Analysis
- Full Restart Analysis

See the *ANSYS LS-DYNA User's Guide* for more information on these analysis types.

3.2.1.2. The File Management Tab

This tab contains the information necessary to manage your files, such as location of your working directory and job name. The available options will differ depending on the simulation environment you selected on the first tab.

If you selected the **ANSYS** or **ANSYS Workbench** simulation environment you can specify:

Working directory
Sets the directory in which the ANSYS run will be executed. The program writes files it generates to this directory. To change the working directory, type the new directory name in the **Working Directory** text box or press the **Browse** button to display a file selection dialog box.

You can also specify the working directory by defining the **ANSYS81_WORKING_DIRECTORY** environment variable. However, the setting defined in the launcher will override the environment variable setting.

If you run ANSYS as an administrator using a particular working directory, and then run as a non-administrator using the same directory, you may encounter permission problems with the **.log** file. Users with administrator privileges should not use the same working directory as non-administrator users.

Jobname
Defines the base filename used for all files generated by the ANSYS run. The initial jobname defaults to **file**. You can change it to any alphanumeric string up to 32 characters long.

If you selected the **ANSYS Batch** simulation environment you can specify the above items as well as the following:

Input file
Specifies the file of ANSYS commands you are submitting for batch execution. You can either type in the desired file name or click on the **Browse** button to display a file selection dialog box or click on Edit to view the file in an editor. If you specify a relative path, it will be relative from the working directory as specified in the launcher.

Output file
Specifies the file to which ANSYS directs text output by the program. If you specify a relative path, it will be relative from the working directory as specified in the launcher. If the file name already exists in the working directory, it will be renamed to **jobname.out.<*timestamp*>** when a new batch job is started. The timestamp is in the format YYYY-MM-DD@hh_mm_ss and *reflects the date that the file was renamed*, not the date the original file was created. You can choose to overwrite the output file rather than rename it by unchecking the **Save Existing Output Files From Batch Runs** item under the **Options** menu on the launcher.

The **Working directory** and **Jobname** options include **Update I/O path** and **Update I/O name** toggles, respectively. If these toggles are checked, the directory path that appears in the **Input file name** and the name that appears in the **Output file name** text boxes will be updated automatically to match the specified **Working directory** and **Jobname**.

Include input listing in output
Includes or excludes the input file listing at the beginning of the output file.

On UNIX systems, if you wish to start your batch job at a later date, click the **Queue** button. You will be prompted to enter a start date and time. To use this option, you must have permission to run the **at** command on UNIX systems, meaning your user name must appear in the **at.allow** file. If that file does not exist, the **at.deny** file is checked to see if access should be denied. If neither file exists, only root may use this feature. If **at.deny** is empty, global usage is permitted.

These files consist of one user name per line and can be modified only by the superuser. The location of the **at.allow** and **at.deny** files on each hardware platform is listed below:

- HP AlphaServer (Compaq), HP, SGI, Sun: **/usr/lib/cron**
- IBM: **/var/adm/cron**
- Intel Linux: **/etc**

If you selected the **LS-DYNA Solver** simulation environment, you can specify the working directory as described above, and the following:

Keyword Input File:
Specify the name of the **file.k** that ANSYS writes. On Windows systems, the file must reside in the working directory.

Restart Dump File:
Specify the name of the small restart input file. On Windows systems, the file must reside in the working directory. For more information, see the **EDSTART** command.

3.2.1.3. The Customization Tab

The settings under this tab allow you to specify detailed settings about your working environment, such as memory settings, parallel/distributed processing settings, custom executables, and additional parameters. The available options will differ depending on the simulation environment you selected on the first tab.

If you selected the **ANSYS**, **ANSYS Workbench**, or **ANSYS Batch** simulation environment you can specify:

Use Custom Memory Settings
Specifies the use of custom memory settings rather than using the default memory model. You must select custom memory settings in order to set the `-m` or `-db` options. See Chapter 19, “Memory Management and Configuration” in the *ANSYS Basic Analysis Guide* for information about memory management.

Total Workspace (MB)
Specifies the amount of memory requested for the ANSYS run. It defaults to 512 megabytes, which is sufficient for most modest-sized models. Choosing a higher number allows more space for solution of large wavefront models, and choosing a lower number allows more concurrent ANSYS runs. This option is not valid unless **Use Custom Memory Settings** is selected.

> *Note* — If a **config81.ans** file exists with its own work space specification (**VIRT_MEM**), the work space value specified on the launcher overrides the value specified in the **config81.ans** file.

Database (MB)
Specifies the portion (in megabytes) of total memory that the database will use. It defaults to 256 megabytes , which is sufficient for most modest-sized models.

Distributed Processing
Select the distributed processing method you want to use. You can choose to use one of the distributed solvers (requires the Parallel Performance for ANSYS add-on be selected). You then select the MPI type. Not all options or types of MPI are available on all platforms. See the *ANSYS Installation and Configuration Guide* for your platform for more information on configuring Parallel Performance for ANSYS.

Custom ANSYS EXE
The launcher provides an option to start a customized ANSYS executable. See *Guide to ANSYS User Programmable Features* for more information on customizing ANSYS.

Additional Parameters
You can use this option to set parameter values at ANSYS start-up.

For example, you can set ANSYS modeling parameters at program start-up using the format `-name` *`value`*. For instance, -rad1 2.825 -rad2 5.675 -thick .25 defines three parameters: rad1=2.825, rad2=5.675, and thick=0.25. The parameter name must be at least two characters long. See the *ANSYS APDL Programmer's Guide* for details about ANSYS parameters.

> *Note* — If a parameter with the same name is defined in the **start81.ans** file and the file is read, the definition in the **start81.ans** file overrides the definition specified in the launcher.

If you selected the **LS-DYNA Solver** simulation environment you can specify the following items:

Memory (words):
Specify the memory to be used, in words. For more information, see the **EDSTART** command.

File Size:
Specify the scale factor for binary file size. For more information, see the **EDSTART** command.

Number of CPUs:
Specify the number of CPUs to be used for shared memory parallel processing.

Enable Consistency Checking:
Specify if consistency is to be forced or not for shared memory parallel processing. Forcing consistency will ensure that calculations are performed in the same order across all machines, avoiding differences in results that could otherwise occur.

Enable Double Precision Analysis:
For more accurate results, you can use the double precision capabilities of ANSYS LS-DYNA. This feature is especially useful in sequential explicit-to-implicit spring back types of analyses.

For more information on these settings or ANSYS LS-DYNA in general, see the *ANSYS LS-DYNA User's Guide*.

3.2.1.4. The Preferences Tab

Use the options here to specify your ANSYS preferences.

If you selected the **ANSYS**, **ANSYS Workbench**, or **ANSYS Batch** simulation environment you can specify:

ANSYS Language
: Specifies a translated language file. If you specify a language that does not have the associated translated files, the launcher defaults to US English. ANSYS, Inc. does not provide translated message files for all products. For information on the availability of translated language files, contact your ANSYS sales representative.

Graphics Device Name
: Sets the graphics device. The GUI requires a terminal that supports graphics. For UNIX systems, the graphics device can be either X11 (default) or 3D. Choose 3D if you have a 3-D graphics device; otherwise, use the default. For Windows systems, WIN32 is the default graphics device. Not available with the ANSYS Batch simulation environment.

Send output to
: (UNIX only) Directs the output of the ANSYS program to the screen only, to an output file only, or to both. For systems that support two-way redirection of output, the default is to send output to both the screen and the output file (**Jobname.OUT**).

Read start.ans at start-up
: Suppresses reading of the **start81.ans** file. You can include commands to be executed when the program starts up in the **start81.ans** file. See Section 3.6: Setting Preferences with the start81.ans File for more information.

3.2.1.5. The LSF Tab

If you selected the **ANSYS/Batch** simulation environment, and you have Platform Computing's Load Sharing Facility (LSF/Batch) program installed, you will see this tab. ANSYS LSF/Batch is not supported on all platforms.

Platform Computing's Load Sharing Facility (LSF/Batch) is a separately-purchased product that supports batch job management and intelligent queues.

When you select the LSF tab, you will see a list of Host Types, Available Hosts, and Available Queues.

The Host Type box lists the type of machines (such as HP PA11) available as hosts at your site. Available Hosts are the actual machine names available for your use on the network. Available Queues are predefined queues that you have permission to access.

You can also specify additional input and output files to be transferred via LSF. To add an additional file, click **New** (either under **Additional Input Files** or **Additional Output Files**) and enter the name of the file in the text entry dialog that appears and press the **OK** button. The file is added to the list. To delete a file, select the file from a list. The file name appears in the text entry box under the list. Highlight the file name and press the **Delete** key.

If you make selections or changes and want to return to your default settings on this tab, click the **Refresh** button.

Select the appropriate values, then choose the **LSF Run** button to send the batch job to a queue for processing.

For information on how to use ANSYS LSF/Batch, check the documentation on this product from Platform Computing. The LSF/Batch product is documented in the manual *LSF JobScheduler User's Guide*, available online and from Platform Computing.

You can alternatively launch LSF/Batch from the ANSYS input window. See Section 3.4.2: Launching LSF/Batch from the ANSYS Input Window.

To configure LSF to run correctly on your system, see Section 3.4.3: Configuring LSF/Batch on Your System

3.2.2. Launcher Menu Options

You can find launcher-specific controls under several launcher menu options. The menu options are:

File

Use the **File** menu to restart the launcher if it is already running or to exit.

Profiles

Use the **Profiles** menu to save a specific launcher configuration. The profile you are currently running is displayed in the launcher title bar. An asterisk (*) follows the profile name if you have changed any settings from what is saved.

- Click **Save Profile** to save the current launcher configurations. You can enter a name for this profile. The last run profile will automatically be set as the default unless you specify otherwise. If you want to set a subsequent profile as the default, you will need to check the **Set as Default** box when you define that profile.
- Click **Load Profile** to select the last run profile, the initial settings (the default configuration as shipped by ANSYS), or one of your saved profiles.
- Click **Manage Profiles** to rename a profile, specify a different default profile, or delete a profile.

Once you've run the launcher at least once and established a “last run” profile, you can quickly “run now” using the most recent launcher configuration by typing:

```
launcher81 -runae
```

`runae` will launch the ANSYS environment. You can launch other simulation environments using the following options instead of **`runae`**:

ANSYS Batch: **`-runbatch`**
ANSYS Workbench: **`-runawe`**
ANSYS LS-DYNA: **`-rundyna`**

When you “run now,“ you can also specify a particular profile using the `-profile` *`profilename`* option:

```
launcher81 -runae -profile myprofile1
```

where *`myprofile1`* is the name of a previously-defined profile. You can specify “default” as the profile name to launch the product using your default profile:

```
launcher81 -runae -profile default
```

Options

Use the **Options** menu to select or deselect the following options:

- Close the launcher on run
- Write out the launcher log file (launcher.81.log)
- Show the hostname in the launcher title bar
- Use alternate working directory browse dialog box
- Pause at the end of a run (where applicable)
- Specify your terminal emulator (UNIX only)
- Save existing output files from batch runs

- Reset ANSYS GUI Configuration (Windows only)

If you choose to use the alternate working directory browse dialog box, it shows you what files are contained in each directory, allows you to create a new directory, and allows you to filter on file type. The standard browse dialog displays only the directories from which to choose.

By default, ANSYS saves existing output files from batch runs. When a new file is created, the old one is renamed to jobname.out.<timestamp> when the batch job is started. The timestamp is in the format YYYY-MM-DD@hh_mm_ss and *reflects the date that the file was renamed*, not the date the original file was created. To overwrite the output file rather than renaming it, uncheck the **Save Existing Output Files From Batch Runs** item under the **Options** menu on the launcher.

Use the Reset ANSYS GUI Configuration option to return the ANSYS GUI to its default configuration if you've previously rearranged the menus or toolbars. This option is especially useful if you've moved an ANSYS GUI component out of your screen's viewable area.

Tools

Use the **Tools** menu to view the ANSYS command line, to open a file in a text editor, or to access ANSYS' auxiliary programs. You may or may not see all of these programs and utilities, depending on what products you have installed. The programs you can access include:

- **ANS_ADMIN** utility
- ANIMATE utility (Windows only)
- DISPLAY utility
- Results Tracker utility. See the **NLHIST** command for more information.
- CFX Launcher. This option is only applicable if you have a CFX product installed. On Windows, this will open the launcher for the last-installed version of CFX. This version will typically be the latest version of CFX you have on your machine. However, if you installed an earlier version of CFX *after* a later version, then this option will open the launcher of that earlier version. On UNIX systems, this option will open the launcher for the version of CFX 5 that appears first in your path.

You can also display your license status, run the **ANSLIC_ADMIN** utility, or view the launcher log file from the **Tools** menu. For more information on the **ANSLIC_ADMIN** utility, see the *ANSYS, Inc. Licensing Guide*.

If you have LSF/Batch installed, you can display the LSF command line as well.

Help

Use the **Help** menu to access more information. From this menu, you can access:

- Launcher help for more detailed information on this launcher.
- Site Information.
- The About Launcher information.

3.3. Interactive Mode

In interactive mode, you work with menus and dialog boxes to drive the ANSYS program. You have easy access to ANSYS graphics capabilities, online help, and other tools, such as wizards.

The standard ANSYS GUI is the default. This layout shows the Utility Menu, Standard Toolbar, Input Window, ANSYS Toolbar, Main Menu, Graphics Window, Status Area, and Output Window. You can resize the toolbars, the overall size of the GUI, the font, and the color. For more information on these components and how to customize them, see Customizing ANSYS and the GUI

You can also run ANSYS through the ANSYS Workbench Products. The ANSYS Workbench provides a framework for integrating the various ANSYS computer-aided engineering tools into a single working environment, combining the user interface strengths of the DesignSpace product with the solution capabilities of ANSYS.

To run the ANSYS Workbench, you must first install the ANSYS Workbench Products. See the *ANSYS Workbench Products Installation and Configuration Guide* for your platform for more information.

3.3.1. Executing the ANSYS or DISPLAY Programs from Windows Explorer

If you are running ANSYS on a Windows system, you can double-click on the following types of files from the Windows Explorer to execute the ANSYS or DISPLAY programs:

- Double-click on a **.db** or **.dbb** file to execute the ANSYS program. When executed in this way, ANSYS will use the **Initial jobname**, **Total Workspace** (**-m**), and **Database** (**-db**) values previously set with the **Configure ANSYS** option. To change these settings, select **Configure ANSYS** from the ANSYS folder, change the settings as desired, and click on Close.
- Double-click on a **.grph** or **.f33** file to execute the DISPLAY program. The first plot that appears in the file will be loaded into DISPLAY automatically.

3.4. Batch Mode

In batch mode, you submit a file of commands to the ANSYS program. On some operating systems, you can run a batch job in the background while doing other work on the computer. Batch mode is useful when you do not need to interact with the program, such as during the solution phase of an analysis.

3.4.1. Starting a Batch Job from the Command Line

Starting Batch Mode from the UNIX Command Line

To start batch mode from the UNIX command line:

Foreground execution (ksh or sh shells):

```
ansys81 -b -p productvar < inputname > outputname 2>&1
```

Background execution (ksh or sh shells):

```
nohup ansys81 -b -p productvar < inputname > outputname 2>&1 &
```

The **nohup** command tells the system to ignore hang-up signals, enabling the ANSYS program to continue executing if you log off from the system.

Foreground execution (csh shell):

```
ansys81 -b -p productvar < inputname > &outputname
```

Background execution (csh shell):

```
nohup ansys81 -b -p productvar < inputname > &outputname &
```

Starting Batch Mode from the Windows Command Line

You can also start a batch job in Windows by issuing the ANSYS execution command directly from the MS-DOS command prompt window. The format for the command depends on whether you want ANSYS to run in the foreground or the background:

Foreground execution:

```
"<drive>:\Program Files\Ansys Inc\V81\ANSYS\bin\<platform>\ansys81" -b -i
     inputname -o outputname
```

To run multiple consecutive jobs on Windows systems, create and run a batch file containing commands similar to the example below:

```
set ANSYS81_PRODUCT=ANE3FL
set ANS_CONSEC=YES
"C:\Program Files\Ansys Inc\V81\ANSYS\bin\intel
     \ansys81" -b -i vm1.dat -o vm1.out
"C:\Program Files\Ansys Inc\V81\ANSYS\bin\intel
     \ansys81" -b -i vm2.dat -o vm2.out
"C:\Program Files\Ansys Inc\V81\ANSYS\bin\intel
     \ansys81" -b -i vm3.dat -o vm3.out
```

Setting **ANS_CONSEC=YES** disables ANSYS dialog boxes so that multiple jobs can run consecutively without waiting for user input.

> *Note* — The example above assumes that the ANSYS product that you are running is ANSYS Multiphysics, which has a product variable of ane3fl.

3.4.2. Launching LSF/Batch from the ANSYS Input Window

LSF/Batch can also be launched from the ANSYS Input window. The command syntax is:

```
~LSFBATCH,<command>,<hostname><queue>,<memory>,<batch input>,<batch output>,
<input file>,<output file>
```

Here is a list of brief descriptions of the command arguments. For more information, see the *LSF JobScheduler User's Guide*.

> *Note* — Any parameter containing either a back slash or a forward slash (\ or /) must be enclosed in single quotes. So if the *<batch input>* file is in another directory, the path to the file would look like this:
>
> ```
> ~LSFBATCH,,,,'/library/models/bridge6.inp'
> ```

<command>
The ANSYS command to run as a batch job; this is a required parameter.

<hostname>
The host to submit the batch job to; the default is unspecified (host chosen by LSF).

<queue>
The queue that the job is to be submitted to.

<memory>
The memory resource requirement, which is *not equivalent* to the `-m` option for ANSYS.

<batch input>
: The input file containing the ANSYS model; this is a required parameter.

<batch output>
: The ANSYS output file information.

<input file>
: Any additional file that might be needed, such as a configuration file, a **.db** file, or a part file for an ANSYS Connection test.

<output file>
: Any output file that you wish to be copied back to the submitting host.

3.4.3. Configuring LSF/Batch on Your System

LSF/Batch requires some configuration by the administrator of your system. If LSF/Batch has been licensed for your system but does not run, the administrator will need to update the following files:

- ***<lsf_install>*/lsf/conf/lsf.shared**
- ***<lsf_install>*/lsf/conf/lsf.cluster.*<clustername>***

Detailed instructions on updating these files appears in the *LSF JobScheduler User's Guide* available from Platform Computing. Below are brief samples showing you the information that needs to be changed for each system on which you wish to run LSF/Batch.

When Platform Computing's LSF/Batch product is installed, it should have one central configuration directory for the entire cluster. The configuration directory contains a number of files, including **lsf.shared** and **lsf.cluster.*<clustername>***. The **lsf.shared** file defines information about the type of software run on the network. The **lsf.cluster.*<clustername>*** file lists each host and defines specific applications for each host.

To run LSF/Batch with ANSYS on the network, the administrator first must add the following line to appropriate location in the **lsf.shared** file:

```
ansys     Boolean ( )          ( )       (Hosts with ANSYS)
```

Once the administrator has defined the resource name, add the following information to the **lsf.cluster.*<clustername>*** file:

```
<hostname>      <model>   <type>   <server>       --   (ansys)
```

Note — For *<server>*, indicate a server by entering 1.

See the *LSF Administrator's Guide*, available from Platform Computing, for detailed information on updating these files.

You must refer to the LSF/Batch documentation from Platform Computing for specific information on installing, configuring, and running LSF/Batch.

For ANSYS LSF/Batch to run correctly, your current working directory must exist on all machines where jobs will be run and must match exactly the current working directory of the machine from which the batch job was submitted.

If the names do not match exactly, LSF/Batch assumes that the current working directory does not exist and sets a different location (**/net/hostname/username**) as the working directory on the target machines. If this second location does not exist, your ANSYS jobs could abort with an output message indicating that no working directory exists:

***** FORTRAN I/O ERROR 908: COULD NOT OPEN FILE SPECIFIED FILE: ftn19, UNIT: 19**

You can alternatively add the line LSF_AM_OPTIONS=AMNEVER to the **lsf.conf** file. This statement directs LSF/Batch to establish the working directory on each machine as it does the home directory (i.e., current, followed by $HOME, followed by **/tmp**). Note that you must manually stop and restart the daemons for the entire cluster in order for any changes to the **lsf.conf** file to take effect because this file is read only at the startup of the daemons. For more information on this file, please refer to the LSF/Batch documentation from Platform Computing.

3.5. Choosing an ANSYS Product

ANSYS builds a variety of products. Your site may have licenses for one, several, or all ANSYS products and product combinations. By invoking ANSYS with the appropriate product variable, you can run a more efficient ANSYS session. You can specify the appropriate product via the launcher or via command line using the `-p` option. The default product (if you are licensed for it) is ANSYS Mechanical. See the Product Variable Table in the *ANSYS, Inc. Licensing Guide* for a complete list of all product variables.

On both Windows and UNIX systems, you can start any ANSYS product you have licensed by including the `-p` option when you issue the ANSYS execution command, as shown below:

```
ansys81 -p productvar
```

The value *productvar* is a variable that represents the ANSYS 8.1 product you wish to run. Each ANSYS product has a unique product variable. For example, to start the ANSYS Mechanical product with the add-on FLOTRAN option, type the following:

```
ansys81 -p ANFL
```

3.5.1. Changing the Default Product for a UNIX System

If you regularly use a product other than ANSYS Mechanical but want to avoid typing in the `-p` option and the appropriate product variable each time you start an ANSYS session, follow the shortcut procedure described below. This procedure changes the default product from ANSYS Mechanical to the product you specify.

1. Using **vi**, **emacs**, or another text editor, insert the following line into your **.cshrc** file. (Type the product variable for the product in place of *`productvar`*.)

   ```
   setenv ANSYS81_PRODUCT productvar
   ```

2. Save your changes to the **.cshrc** file, then issue this command:

   ```
   source .cshrc
   ```

Since the product you added to the **.cshrc** file is now the default product, you can omit the `-p` option when you want to run that product. However, to invoke any execution command options other than `-p`, you still need to include them in the command text.

> *Note* — The ANSYS product you specify with the `-p` option does not affect the ANSYS product choices available to you in the launcher. When you invoke the launcher, choose the product to run from the launcher's product menu, as described in Section 3.2.1.1: The Launch Tab instead of using the `-p` option.

You can also set the **ANSYS81_ALTPRODS** environment variable to specify alternate product choices with the priority you wish. See the *ANSYS Installation and Configuration Guide* for your platform for more information on this and other environment variables.

3.5.2. Changing the Default Product for Start-up on Windows

If you customarily use a product other than ANSYS Mechanical but want to avoid typing in the `-p` option and the appropriate product variable each time you start an ANSYS session, follow the procedure described below:

1. Click on the **System** icon in the **Control Panel**.
2. Click on the **Environment** tab.
3. In the dialog box, under **System Environment Variables**, add the **ANSYS81_PRODUCT** environment variable.
4. Set the value of the **ANSYS81_PRODUCT** environment variable to *productvar*.
5. Click on **Set**.

```
set ANSYS81_PRODUCT=productvar
```

The product you chose as the default starts automatically when you issue the ANSYS execution command, and you can omit the `-p` option. To invoke any command options other than `-p`, you need to include them in the command text.

You can also set the **ANSYS81_ALTPRODS** environment variable to specify alternate product choices with the priority you wish. See the *ANSYS Installation and Configuration Guide* for your platform for more information on this and other environment variables.

3.6. Setting Preferences with the start81.ans File

A sample ANSYS start-up file, **start81.ans**, is included in the ANSYS apdl directory (for example, **/ansys_inc/v81/ansys/apdl/start81.ans** on UNIX systems, **Program Files\Ansys Inc\V81\ANSYS\apdl\start81.ans** on Windows systems). Copy this file to your home directory, where you can modify the file to suit your preferences.

You can specify commands to be executed at program start-up in the **start81.ans** file. For example, if you use certain functions frequently during an ANSYS session, you might define them as abbreviations (see Section 2.14.4: Abbreviations) and define those abbreviations in the **start81.ans** file.

By default, ANSYS reads **start81.ans** at the beginning of an interactive session but does not read it during a batch session. Change these defaults using the `-s` option when executing the ANSYS command. If you issue the following command, the **start81.ans** file will not be read even during an interactive session:

```
ansys81 -s noread
```

Conversely, issuing the command shown below forces the program to read the **start81.ans** file during batch execution:

```
ansys81 -b -s
```

You also can specify whether the **start81.ans** file is to be read by using the Interactive or batch dialog box on the launcher. ANSYS reads the first **start81.ans** file it finds in the following search paths:

- The working directory
- Your home directory
- The ANSYS apdl directory

Additional values for the `-s` option are detailed in the following section.

3.6.1. The start81.ans File

Changes made in the **start81.ans** file, especially those concerning hardware or system configuration, could affect other programs after your ANSYS session is completed. Also, ANSYS configuration values that are set during an ANSYS session can carry over into the next ANSYS session if they are not reset. You can reset system parameters and ANSYS configuration values by including a **stop81.ans** file in the same directory as the **start81.ans** file.

By default, ANSYS reads **start81.ans** and **stop81.ans** at the beginning and end of an interactive session, respectively. You can change these defaults by using the `-s` option when executing the ANSYS command. The following values are valid for use in conjunction with the `-s` option:

`-s` ***Value***	**Read start81.ans?**	**Read stop81.ans?**
[default]	Yes	Yes
Noread	No	No
Nostart	No	Yes
Nostop	Yes	No

Both the UNIX and Windows launchers allow you to specify whether or not to read the **start81.ans** file when the program is started. The value you choose for the **start81.ans** file is also applied to the **stop81.ans** file.

3.7. Estimating ANSYS Run Time

The ANSYS program contains a run time estimator that can help you predict the length of ANSYS runs. The estimator uses an ANSYS macro file called **SETSPEED**, which contains the ANSYS command **RSPEED** and an estimate of the speed of the computer on which the macro was created.

The **SETSPEED** macro is created by a stand-alone program called ANSSPD, supplied by ANSYS. The ANSSPD program gathers your computer's performance information and writes it to the **SETSPEED** macro.

3.7.1. Creating a SETSPEED Macro File

You must run the ANSSPD program to create the **SETSPEED** macro file.

On UNIX systems, issue the following command from command level to create the **SETSPEED** macro:

```
/ansys_inc/v81/ansys/bin/ansspd81
```

On Windows systems, issue the following command from the DOS prompt:

```
Program Files\Ansys Inc\V81\ANSYS\bin\<platform>\ansspd
```

ANSSPD creates a **SETSPEED.MAC** file in your current directory. To make it available to all ANSYS users, move it to the **apdl** subdirectory.

3.7.2. Using a SETSPEED Macro to Estimate Run Time

To estimate ANSYS run time, issue the ANSYS command **/RUNST** (**Main Menu> Run-Time Stats**). Then issue the commands **SETSPEED** (to tell ANSYS the speed of the system) and **RTIMST** (**Main Menu> Run-Time Stats> Individual Stats** -- to invoke the run time estimator).

If you are running the ANSYS program on more than one computer, and those computers have different speeds, you may want to run the **SETSPEED** macro on each of your systems. If so, rename the **SETSPEED** file to identify the file uniquely for each system. (For example, you could rename the file as **SPD500.MAC** for a 500 MHz PC, or as **SPDRS6.MAC** for an IBM RS/6000.) Remember to notify all ANSYS users of any filename changes. So if you run **SETSPEED** and get unexpected results, you may be running a **SETSPEED** that was created on one computer and exported to another. You can create a new **SETSPEED** macro by running ANSSPD and moving the resulting **SETSPEED.MAC** file to a local directory.

You also may want to create a **SETSPEED** macro file during the day, when the system load is heavy, and another one with a different name at night, when the system load is light. This will produce two different **SETSPEED** macro files, enabling users to estimate how long their ANSYS jobs will run at a given time of day.

If you do create the two macro files, rename them **DAYSPEED.MAC** and **NITESPEED.MAC** and inform all ANSYS users of the change. Store macro files for all ANSYS users in the ANSYS **apdl** subdirectory.

For more information about **/RUNST**, **RSPEED**, and **RTIMST**, see the *ANSYS Commands Reference*.

Chapter 4: Using the ANSYS GUI

4.1. What Is the ANSYS GUI?

The Graphical User Interface (GUI) is an easy and intuitive way to communicate with ANSYS. The GUI is a menu system that lets beginners and experienced ANSYS users perform virtually all ANSYS operations interactively. The GUI mode is often referred to as "interactive mode" in the documentation and the help system.

You use the GUI to communicate with ANSYS interactively. Each GUI interaction manipulates ANSYS commands to perform the operation. Most of the tasks you use ANSYS for can be performed either interactively, or by inputting the appropriate commands (individually or as a batch file. In both cases, all of the commands are recorded in the input history file (**Jobname.LOG**). The GUI allows you to perform an analysis with little or no knowledge of the ANSYS commands, while still adhering to the command level operations.

If you are new to GUI applications, take some time to read Section 4.2: GUI Controls. It explains some basic GUI standards. If you are familiar with other software packages, skip the next section and go on to Section 4.3: Activating the GUI.

4.2. GUI Controls

This section describes the basic GUI components.

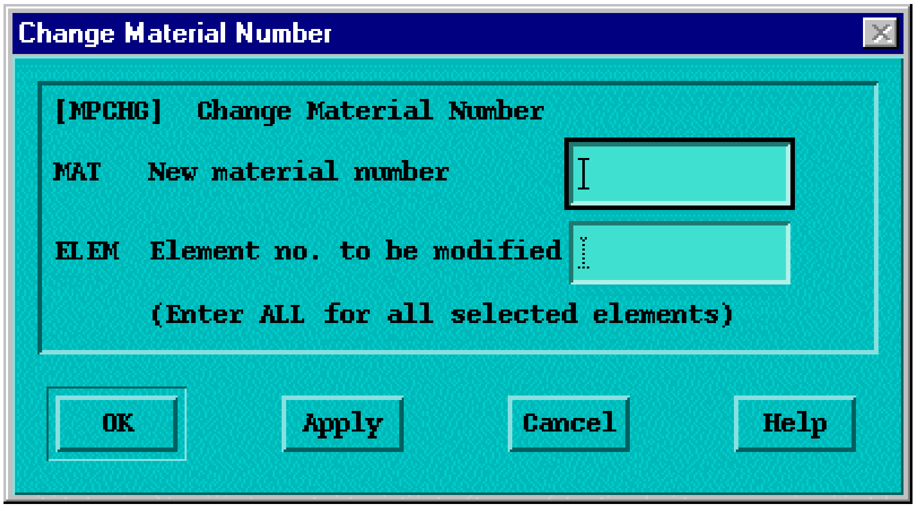
4.2.1. A Dialog Box and Its Components

You use Dialog boxes to provide input to ANSYS for a particular function. The type of dialog box ANSYS provides is dependent on the type of input required. Your input might require any of the following: text entry box, check button, radio button, option button, single-selection list, multiple-selection list, two-column selection list, a tabbed box, or a tree structure. Other components of a dialog box include action buttons such as **OK**, **Apply**, and **Cancel**.

4.2.1.1. Using Text Entry Boxes

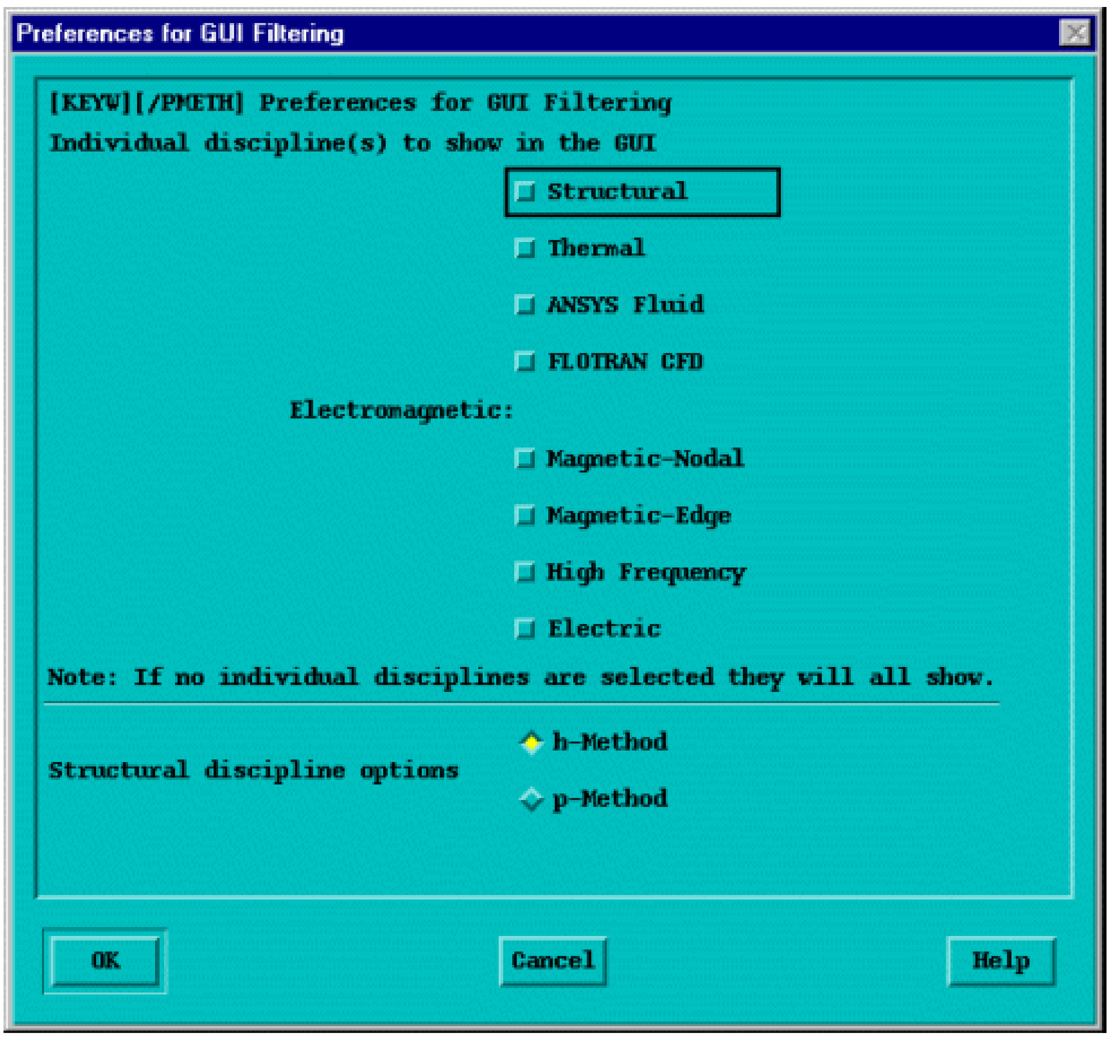
In a text entry box, you enter numbers or character strings (such as parameter names). If text already exists in the text entry box, you overwrite it by highlighting the existing text. To highlight existing text, press and drag the left mouse button. Double-clicking the left mouse button on a word highlights the word, and triple-clicking it highlights the entire string. Figure 4.1: "Text Entry Box" shows an example of a text entry box

Figure 4.1 Text Entry Box



4.2.1.2. Using Check Buttons

Figure 4.2 Check and Radio Buttons



Check buttons are square buttons that turn ANSYS features on or off. To change a check button from on to off or vice versa, you click on it with your left mouse button. In Figure 4.2: "Check and Radio Buttons", the buttons under "Preferences for GUI Filtering" are check buttons.

4.2.1.3. Using Radio Buttons

Radio buttons are diamond shaped buttons you use to choose one of several options. One button is always "On" in a set of radio buttons. Clicking on the desired item makes that button active and turns off other buttons. In Figure 4.2: "Check and Radio Buttons", the buttons shown under "Methodology" are radio buttons.

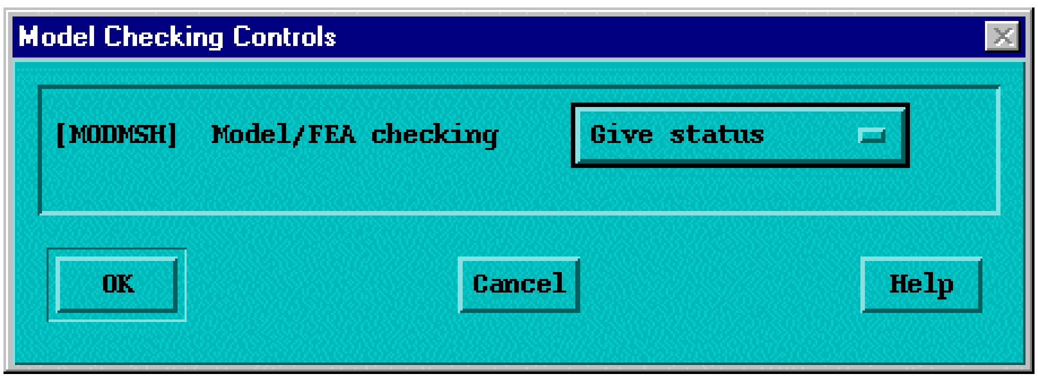
4.2.1.4. Using Option Buttons

With option buttons you choose an item from a pop-up menu that collapses into one button showing the active choice. To choose a different option, do the following:

1. Place the cursor on the button.
2. Press the left mouse button and hold it.
3. Drag the mouse through the pop-up menu until the desired item is highlighted.

Figure 4.3: "Option Buttons" shows a dialog box containing an option button.

Figure 4.3 Option Buttons

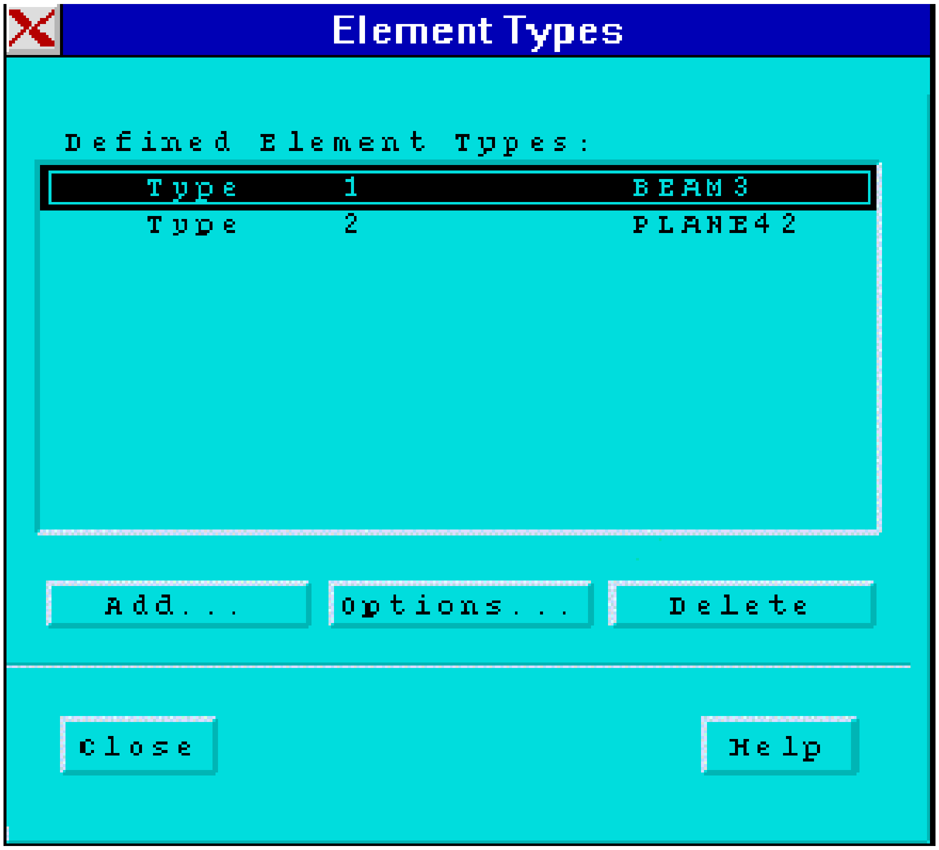


4.2.1.5. Using Single-Selection Lists

You use a single-selection list to choose one option from a scrollable list. Clicking on the desired item highlights it and copies it to the Selection box, where you can edit it.

Figure 4.4: "Example of a Single-Selection List" shows an example of a single-selection list.

Figure 4.4 Example of a Single-Selection List

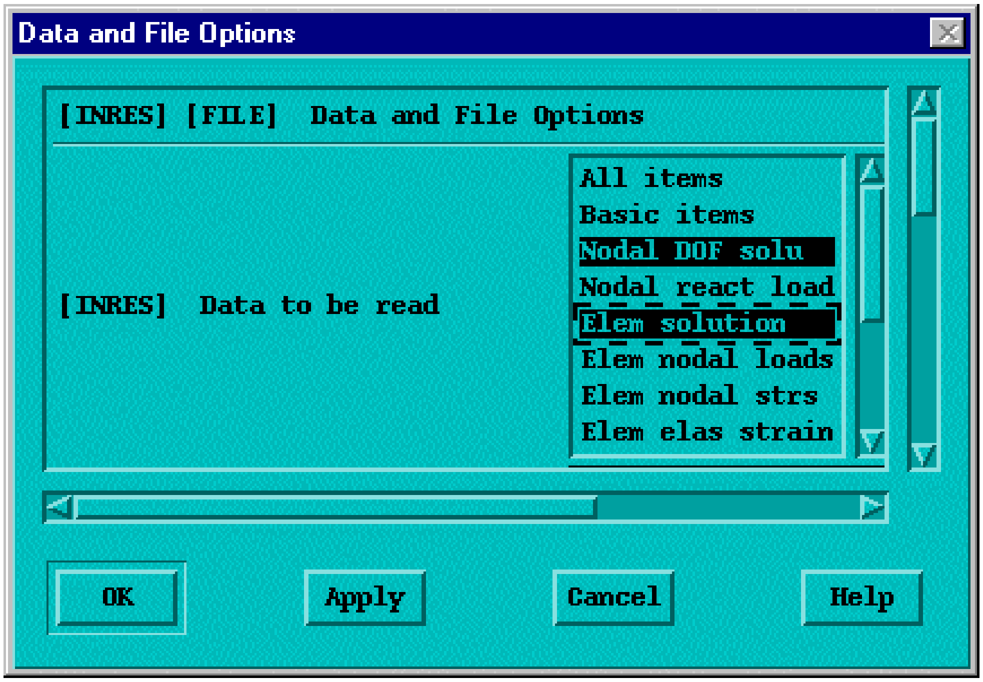


4.2.1.6. Using Multiple-Selection Lists

You use a multiple-selection to choose items from a list. It resembles the single-selection list except that there is no Selection box, and you can choose more than one item. Clicking on an item highlights and makes it active, and clicking the same item again deactivates it.

Figure 4.5: "Multiple-Selection List" shows an example of a multiple-selection list.

Figure 4.5 Multiple-Selection List



4.2.1.7. Using Two-Column Selection Lists

You can select one of several choices from a two-column selection list. It resembles the single-selection list except your choices are grouped into categories. Choose a category from the left column, then pick the desired item from the right column. Choices available in the right column vary according your choice in the left column.

The GUI displays this type of list primarily when you need to choose from a large number of items.

Figure 4.6 Two-Column Selection List

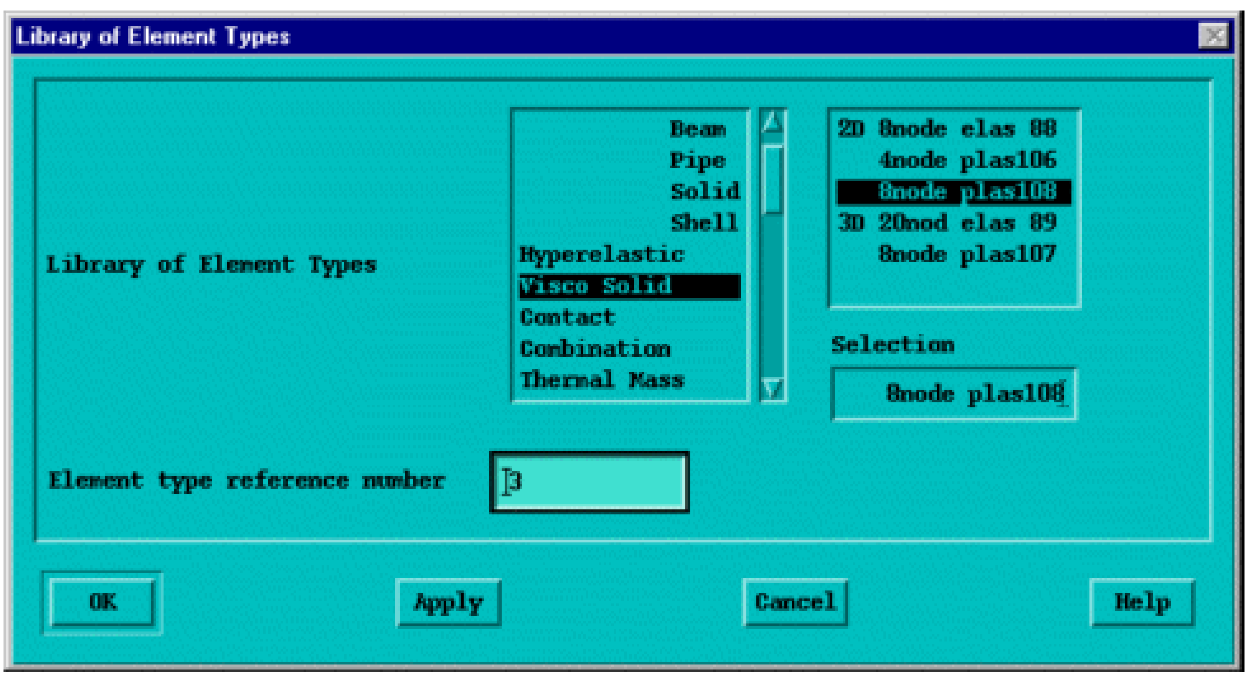
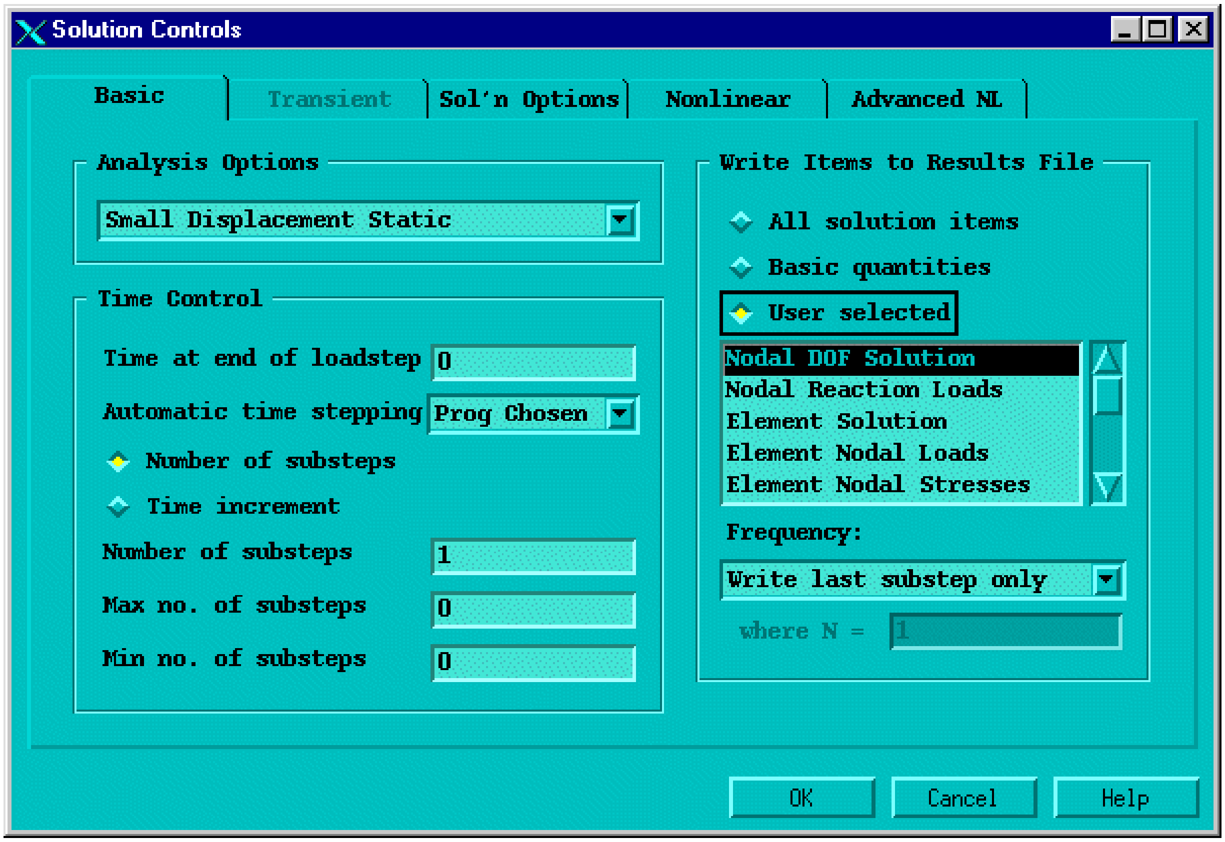


Figure 4.6: "Two-Column Selection List" shows an example of a two-column selection list.

4.2.1.8. Using Tabbed Dialog Boxes

A tabbed dialog box presents groups of related commands in one location. You choose an individual tab by clicking on it, or by using CTRL-Tab to move between the tabs. Some tabbed dialog boxes require that you navigate through each tab sequentially, with later tabs “grayed out” until you complete required tasks in an earlier tab.

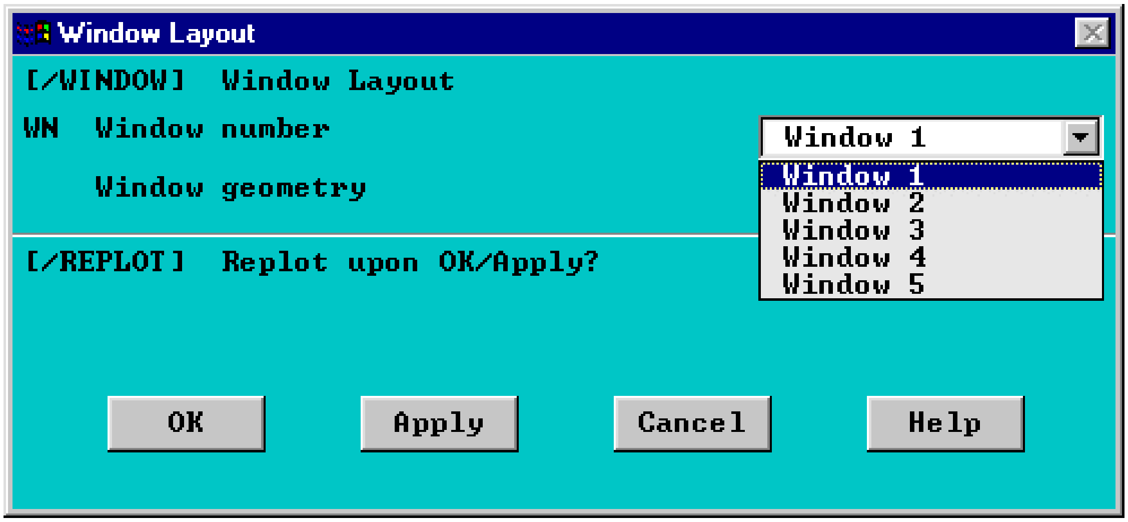
Figure 4.7 Tabbed Dialog Box



4.2.1.9. Using Drop-Down List Boxes

A drop-down list box provides a list of choices for an option. Make a selection by clicking on the small arrow on the right side of the list box and then scrolling to and clicking on the desired choice from the list.

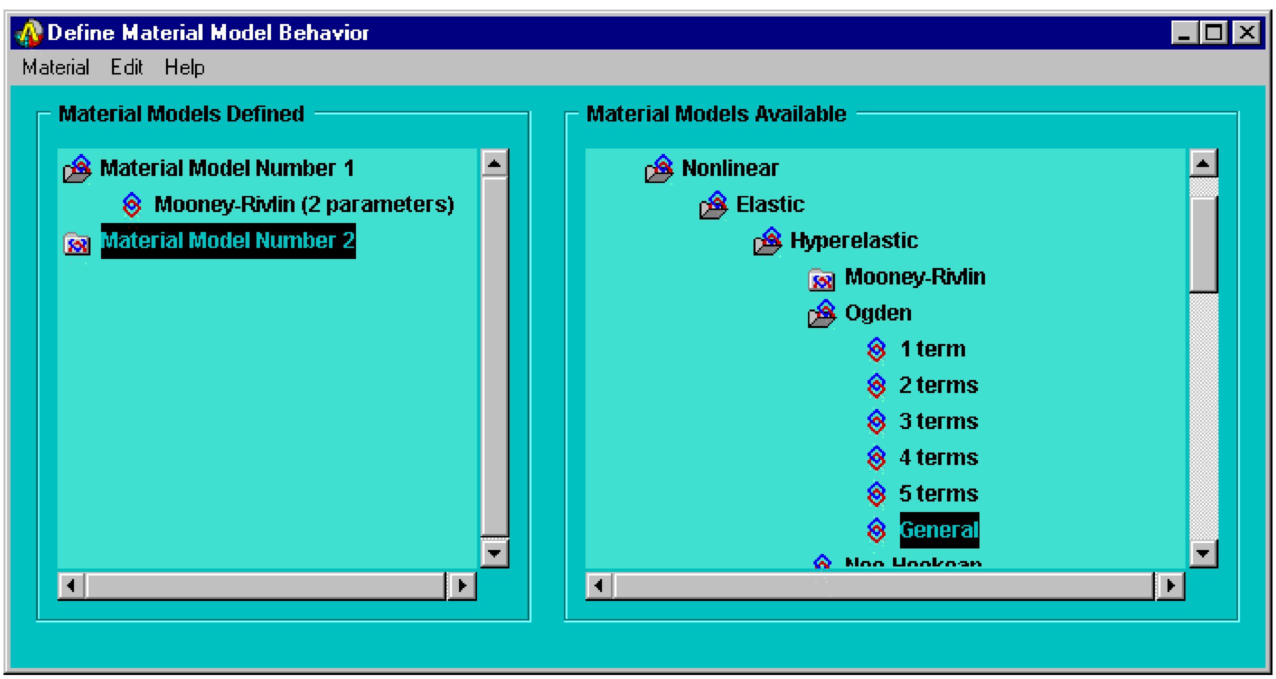
Figure 4.8 Drop-Down List Box



4.2.1.10. Using Tree Structures

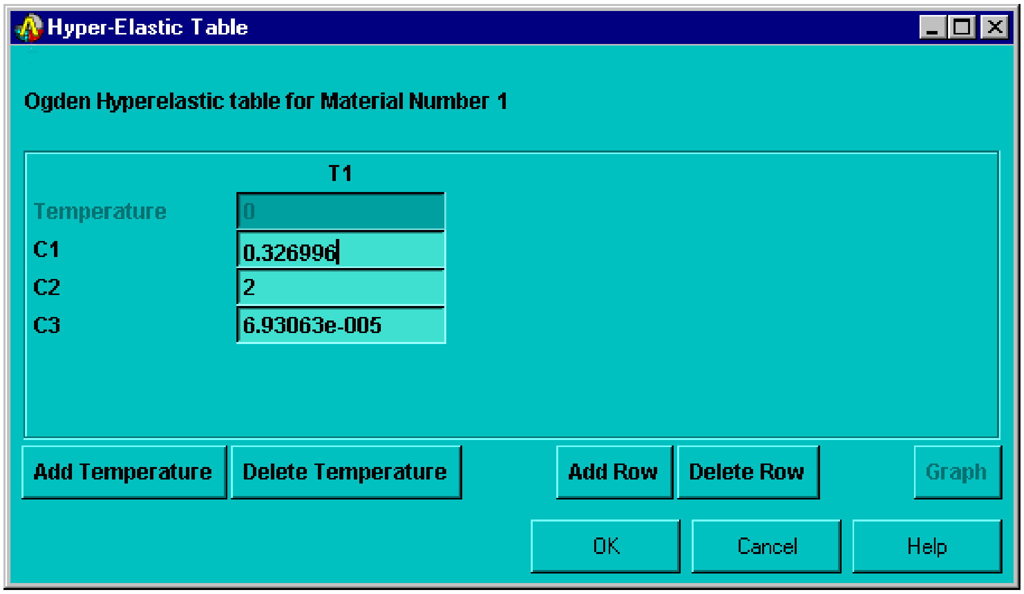
A tree structure presents a hierarchical flow of logical choices from major categories to specific subcategories. Select a major category by double-clicking on the category. Subcategories appear listed vertically and indented beneath that major category. If you select a subcategory in the same way, further subcategories are listed, and become more specific. Continually selecting subcategories yields the final item in the "branch" of the tree. At this point, instead of a categorical choice, you enter or edit data to provide specific input.

Figure 4.9 Tree Structures



You double-click on the name of the final item to display a data input dialog box where you enter or edit the data.

Figure 4.10 Sample Data Input Dialog Box



4.2.1.11. Using Action Buttons

A dialog box typically contains combinations of the following action buttons:

- **OK** - Applies the changes and closes the dialog box.
- **Apply** - Applies changes but *does not* close the dialog box. Use this button when you need to make more than one change within a dialog box, or you need to execute a function more than once.
- **Reset** - The settings in the dialog box revert to the previous or default settings.
- **Cancel** - Closes the dialog box without applying any changes you made. The difference between **Cancel** and **Reset** is that **Reset** does not close the dialog box.
- **Help** - Displays help information for the function being performed.

4.2.1.12. Entering a Mathematical Expressions in a Field

In some cases, you may want to specify an input value in the form of a mathematical expression. To do so, use parentheses around individual operands in the expression, as shown in this example:

```
(-111.5)+(68)
```

If you omit parentheses, ANSYS may interpret the expression differently than you had intended, causing unpredictable results or error messages. For more information about the order in which the ANSYS program evaluates an expression, see Section 3.8: Parametric Expressions.

4.3. Activating the GUI

If you enter ANSYS through the launcher (see Chapter 3, "Running the ANSYS Program"), the GUI is activated *automatically* for interactive running. If you enter the program by typing the ANSYS execution command, you can activate the GUI in any of the following ways:

- Use the GUI entry option (-g) on the execution command. For example, on UNIX systems you would issue the following execution command:

  ```
  ansys81 -g
  ```

- Type **/MENU**,ON once you enter the program.
- Include **/MENU**,ON in your **start81.ans** file.
- Set the **GRPH_ENTR** keyword in your **config81.ans** file. For more information about this file, see Chapter 18, “File Management and Files” in the *ANSYS Basic Analysis Guide*.

For the last three methods listed above, you must first specify a valid graphics device name so that the ANSYS program can properly direct graphics instructions to your display device. To specify a graphics device name, issue the **/SHOW** command.

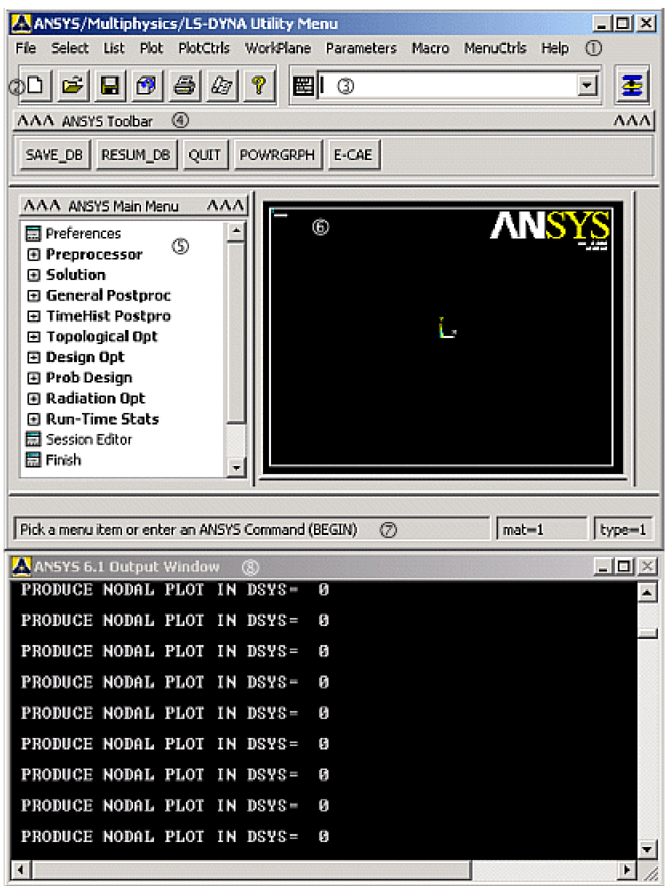
> *Note* — Once you activate the GUI, you cannot deactivate it within the same ANSYS session.

> *Note* — To return to the pre-ANSYS 6.1 GUI, enter **/MSTART**, UTIL,1 in your start81.ans file. You must use pre-ANSYS 6.1 GUI documentation if you choose this option.

4.4. Layout of the GUI

For all ANSYS products except ANSYS Professional, by default the ANSYS GUI has eight areas, as shown in Figure 4.11: “The ANSYS GUI”

Figure 4.11 The ANSYS GUI



1. **Utility Menu** - Contains utility functions that are available throughout the ANSYS session, such as file controls, selecting, graphics controls, and parameters. You also exit the ANSYS program through this menu. See Section 4.4.1: The Utility Menu for more information.

2. **Standard Toolbar** - Contains graphic buttons that execute frequently used ANSYS commands. See Section 4.4.2: The Standard Toolbar
3. **Command InputArea** -Allows you to type in commands directly. The Single Line Input Window (shown) is the GUI default, but an editable window is available for more complex command operations. See Section 4.4.3: Command Input Options for more information.
4. **ANSYS Toolbar** - Customizable toolbar that contains push buttons that execute commonly used ANSYS commands and functions. You may add your own push buttons by defining abbreviations. See Section 4.4.4: The ANSYS Toolbar for more information.
5. **Main Menu** - Contains the primary ANSYS functions, organized by processors (preprocessor, solution, general postprocessor, design optimizer, etc.). See Section 4.4.5: The Main Menu for more information.
6. **Graphics Window** - A window where graphics displays are drawn. See Section 4.4.6: The Graphics Window for more information.
7. **Status and Prompt Area** - located at the bottom of the GUI, shows prompts and the status of your analysis. Typically, you will see prompts for functions that involve graphical picking. *Be sure to read the prompt so you can pick the proper entities in the proper order.* This area also contains status information (PREP7, SOLU, etc.).
8. **Output Window** - Receives text output from the program. It is usually positioned behind the GUI, but you can bring it to the front when necessary. See Section 4.4.7: The Output Window for more information.

You can resize the ANSYS toolbar, Main Menu and Graphics Window, as well as the overall size of the GUI. To resize the areas in the GUI, drag the borders around the areas of the GUI while holding down the left mouse button.

To change the overall size of the GUI, position the mouse on of the corners of the GUI and drag it diagonally towards the center of the GUI while holding down the left mouse button. You can save your GUI size settings by selecting **Utility Menu> MenuCtrls> Save Menu Layout**.

To completely remove ANSYS from your screen without terminating the ANSYS session, simply iconify the GUI. Later, you can bring back the ANSYS session by restoring the icon.

The remainder of this chapter describes each of the areas of the GUI.

4.4.1. The Utility Menu

The Utility Menu contains ANSYS utility functions such as file controls, selecting, graphics controls, and parameters. You can execute most of these functions anytime during the ANSYS session.

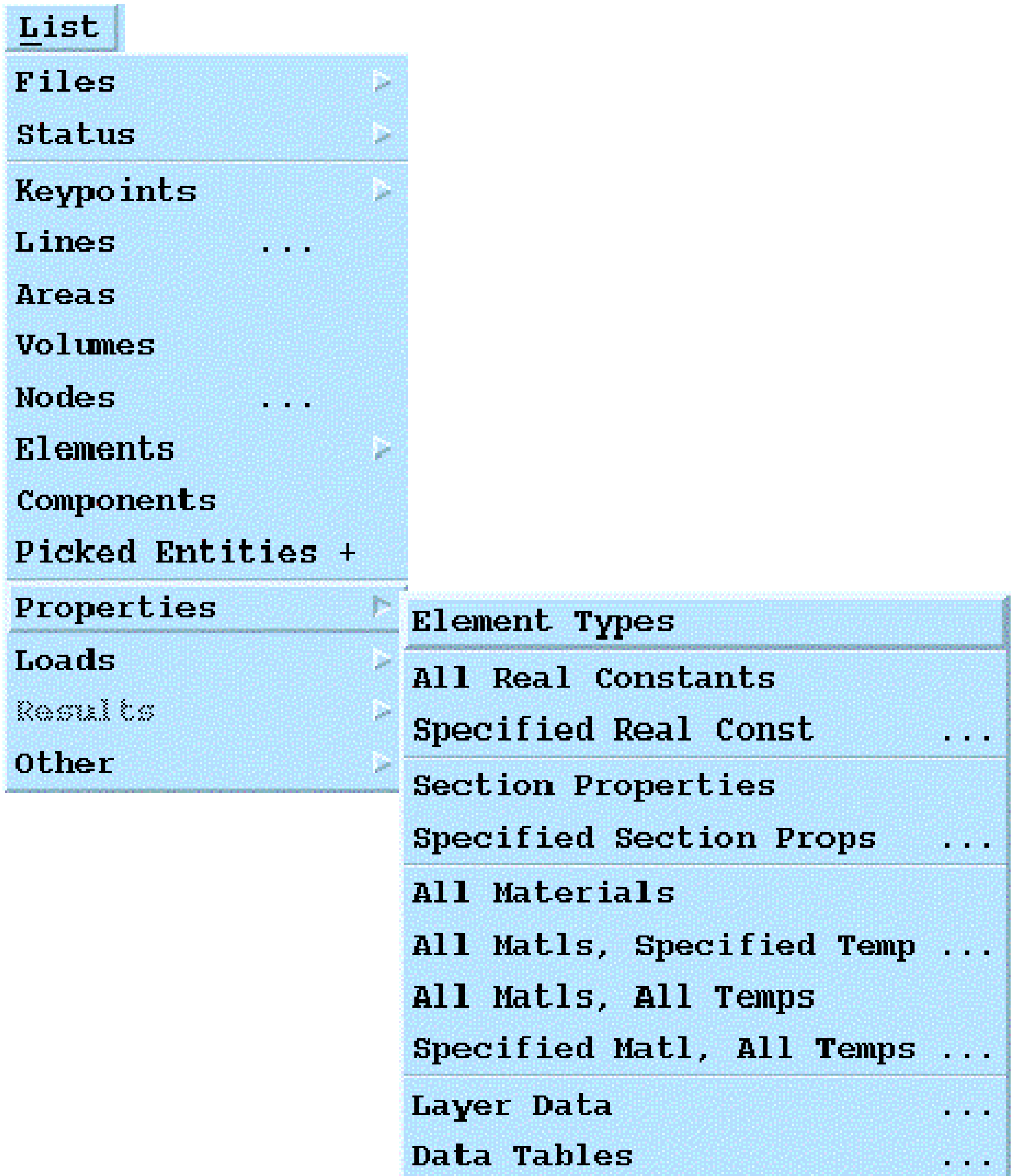
For example, while picking locations on the working plane to create keypoints (a Main Menu function), you can choose **Utility Menu> PlotCtrls> Pan-Zoom-Rotate> Iso** to change the view to an isometric view. If this function were "modal," then you would have to complete the "Create Keypoints" operation before being able to change views.

Each menu topic on the Utility Menu brings up a pull-down menu of subtopics, which in turn either cascade to a submenu (indicated by a **>**) or perform an action. The action may do any of the following:

- Immediately execute a function.
- Bring up a dialog box (indicated by a **...**).
- Bring up a picking menu (indicated by a **+**).

Figure 4.12: “Example of a Pull-Down, Cascading Menu” shows the pull-down list of subtopics you see when you choose the **List** topic:

Figure 4.12 Example of a Pull-Down, Cascading Menu



You use the left mouse button to "pull down" a menu topic on the Utility Menu. Pressing and dragging the mouse button allows you to move rapidly to the desired subtopic. Releasing the mouse button while it is on an "action" subtopic causes ANSYS to perform that action. *Clicking* the left mouse button leaves the pull-down and cascading menus in place. The menus disappear when you click on an action subtopic or elsewhere in the GUI.

You can also use your keyboard to pull down a menu topic and move to the desired function. You do so via the menu's *mnemonic* character, indicated by an underscore. Simply place the mouse cursor anywhere in the Utility Menu and press the ALT key and the mnemonic character simultaneously. For example, pressing ALT+F with the mouse cursor anywhere in the Utility Menu pulls down the File menu, since F is the mnemonic (indicated by the underscore in **F**ile).

Once the pull-down menu appears, you can use the other mnemonic characters (without the ALT) or the arrow keys on your keyboard to navigate to the desired menu topic and the ENTER or RETURN key to "pick" it.

The Utility Menu lists 10 topics. A brief description of each topic follows.

- **F**ile - Contains file and database related functions, such as clearing the database, saving it to a file, and resuming it from a file. Some of the functions under the **File** menu are valid at Begin level only. If you choose such a function when you are not at Begin level, you will see a dialog box giving you a choice of moving to Begin level and executing the function or canceling the function.
- **S**elect - Includes functions that allow you to select subsets of entities and to create components. See Chapter 7, “Selecting and Components” in the *ANSYS Basic Analysis Guide* for details on selecting and components.
- **L**ist - Enables you to list virtually any data item stored in the ANSYS database. You can also obtain status information about different areas of the program and list the contents of files residing on your system.
- **P**lot - Lets you plot keypoints, lines, areas, volumes, nodes, elements, and other data that can be graphically displayed.
- Plot**C**trls - Includes functions which control the view, style, and other characteristics of graphics displays. The **Hard Copy** function lets you obtain hard copies of the entire screen or just the Graphics Window.
- **W**orkPlane - Enables you to toggle the working plane on or off and to move, rotate, and otherwise maneuver the working plane. You can also create, delete, and switch coordinate systems by using this menu. See the *ANSYS Modeling and Meshing Guide* for details about working planes and coordinate systems.
- Pa**r**ameters - Includes functions to define, edit, and delete scalar and array parameters. See the *ANSYS Modeling and Meshing Guide* for details about parameters.
- **M**acro - Allows you to execute macros and data blocks. You can also create, edit, and delete abbreviations, which appear as push buttons on the Toolbar. The *ANSYS Modeling and Meshing Guide* describes macros, and the Toolbar is discussed later in this chapter.
- Me**n**uCtrls - Lets you create, edit, and delete abbreviations on the ANSYS Toolbar (see Section 4.4.4: The ANSYS Toolbar), and modify the colors and fonts used in the GUI display. Once you've adjusted the GUI to your liking, you can use the **Save Menu Layout** function to save the current GUI configuration (including menu, window and overall GUI size).
- **H**elp - Brings up the ANSYS Help System, described in Chapter 7, “Using the Online Help and Manuals”.

4.4.2. The Standard Toolbar

The Standard Toolbar contains a set of icon buttons that execute commonly used functions (see Figure 4.13: “Standard Toolbar”).

Figure 4.13 Standard Toolbar

By default, the Standard Toolbar is loaded and positioned when you start ANSYS. The Standard Toolbar is defined at start up, and cannot be modified during your session. For information, on modifying, repositioning, or defining additional toolbars see Section 4.4.8: Creating, Modifying and Positioning Toolbars, below.

The standard buttons and their functions include:

New Analysis
Saves and clears information for the existing analysis and starts a new analysis.

Open ANSYS File
Reads ANSYS database or input files to be read into ANSYS. The file type determines the operation.

Save Analysis
Saves the current analysis to a database file.

Pan-Zoom-Rotate
Opens the Pan-Zoom-Rotate dialog box. See Section 9.4: Changing the Viewing Angle, Zooming, and Panning in the *ANSYS Basic Analysis Guide*.

Image Capture
Opens the image capture GUI. See Section 5.4.3.1.2: The Results Viewer Toolbar in the *ANSYS Basic Analysis Guide* for more information.

Report Generator
Opens the report generator GUI. See Chapter 17, "The Report Generator" in the *ANSYS Basic Analysis Guide* for more information on generating reports.

ANSYS Help
Displays the table of contents for the ANSYS HTML-based help.

Raise Hidden
Raises hidden windows to the top of the application.

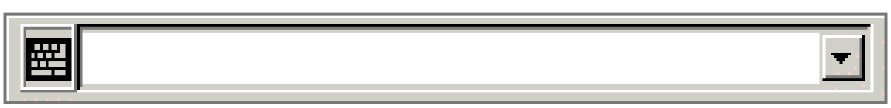
Contact Manager
Opens the Contact Manager GUI.

4.4.3. Command Input Options

Although the GUI provides intuitive graphical access, you can provide input to the program by typing in commands, even while the GUI is active. There are two modes available for directly entering commands during your analysis. The Single Line Input Window displays only one line and uses a drop-down window to display the command history. Or you can use the ANSYS Command Window for more intensive command operations, such as pasting in multiple command strings or copying longer strings and selected commands from the history window. Both modes are described below.

4.4.3.1. The Single Line Input Window

Figure 4.14 Single Line Input Window



The Input Window is the default display for all GUI operations. You use the Input Window to conveniently enter single commands and access the history buffer without changing the overall configuration of the GUI. As you enter commands into the Input Window, dynamic command help appears in a box above the window. As you type the letters, the command help displays the possible commands, and guides you through the proper spelling and syntax of the command.

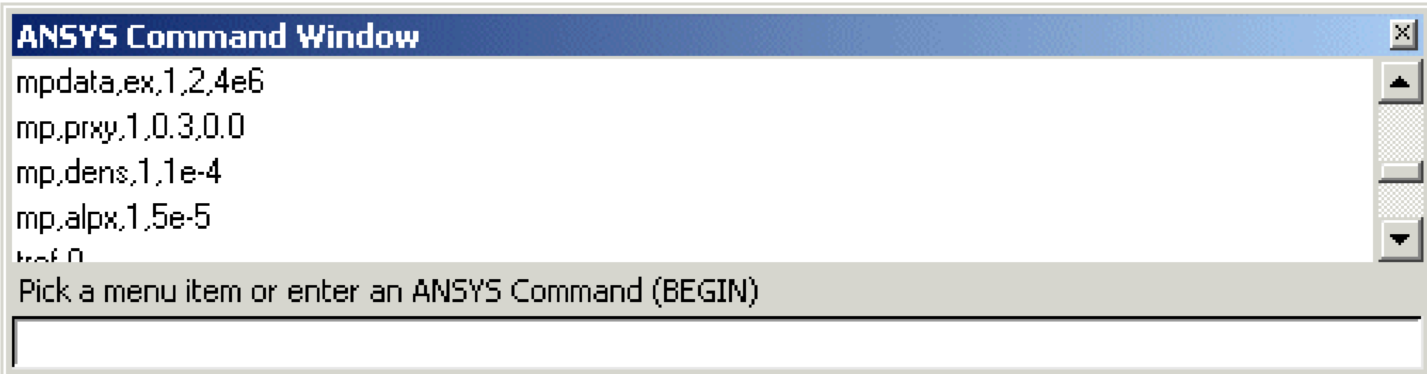
The *history buffer* contains all previously entered commands. Individual lines from the history buffer can be displayed in the Input Window and indexed with the up or down arrow keyboard keys. Once displayed in the window, you can edit the entry and execute it by using the return key.

You can view and access the history buffer by clicking the down arrow on the right of the text entry box. A drop down list containing the entry history appears. Clicking the left mouse button on any line in the history buffer moves that line to the text entry box where you can edit it and execute it. A double click on any line in the history buffer automatically executes that line.

The *vertical scroll bar* at the right corner of the (unhighlighted) history buffer box allows you to scroll through the history buffer. You can also use the up and down arrow keys to navigate the history buffer.

4.4.3.2. The Floating ANSYS Command Window

Figure 4.15 The Floating ANSYS Command Window



You bring the Floating ANSYS Command Window up by clicking on the keyboard icon at the left side of the Single Line Input Window. The ANSYS Command Window is a floating window that can be resized and positioned easily for more complex command entry operations. You can work in the command window, and when you close it, all of your operations will appear in the history buffer.

When you access the command history with the command window, you choose commands from the buffer, and selectively enter them into input line, in any order you choose. You do this by holding down the Ctrl-key and clicking on each command. As you do, the commands appear in the input line below. You can then edit the commands, changing parameters or syntax, before hitting enter to execute them.

You can also set the size and location of the Floating ANSYS Command Window to the desired orientation, and then use the “Save Menu Layout” feature to make it your default (the position and size you designate will appear the next time you start ANSYS). The Single Line Window will be available when you close the Command Window, but will no longer be the default.

The ANSYS Command Window provides a scrollable, editable text area where you can enter individual commands or paste in long command strings. You can also select multiple commands from the history buffer area, either using the shift key to select large contiguous portions of the history, or the control key to select multiple, individual commands. Like the input window, dynamic command help is available; as you type your command, the probable command is displayed. This display shows the complete command syntax, and stays visible so that you can check the help system or check other command argument parameters.

4.4.4. The ANSYS Toolbar

The ANSYS Toolbar is a convenient area where you can add push-button for command, function and macro shortcuts. The Toolbar is a set of push buttons that execute commonly used ANSYS functions. You can set the toolbar up to provide one-button access to your favorite and frequently used functions. It is loaded and positioned by default, and can be modified during your ANSYS session. Some push buttons (for example, **SAVE_DB** and **RESUM_DB**) are predefined, but you define all others. That is, you choose how many push buttons the Toolbar contains (to a maximum of 100) and what functions they execute.

Figure 4.16 The Toolbar

4.4.4.1. Adding Buttons to the Toolbar

You add additional push buttons to the Toolbar by creating abbreviations. An *abbreviation* is simply an alias (up to 32 characters long) for a complete ANSYS command or GUI function name. For example, **SAVE_DB** is an alias for the **SAVE** command, **RESUM_DB** is an alias for the **RESUME** command, and **QUIT** is an alias for the function **Fnc_/EXIT**, which displays the "Exit from ANSYS" dialog box. You can also add *macros* to the Toolbar by defining an abbreviation that executes the macro.

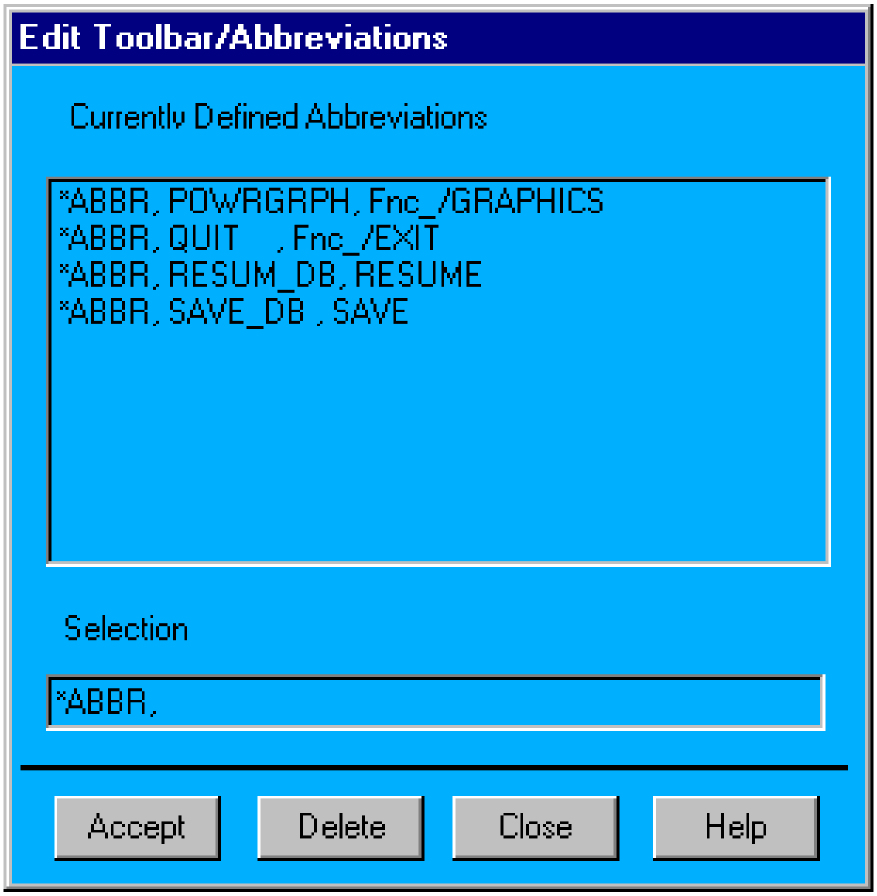
4.4.4.2. Creating Abbreviations

To create an abbreviation, choose **Utility Menu> MenuCtrls> Edit Toolbar** or **Macro> Edit Abbreviations**. Both menu choices bring up the "Edit Toolbar/Abbreviations" dialog box shown in Figure 4.17: “Edit Toolbar / Abbreviations Dialog Box”. The Toolbar immediately reflects any changes you make to the abbreviations using this dialog box.

You can also create an abbreviation by issuing the ***ABBR** command from the Input Window. If you use this method, you need to update the Toolbar manually by picking **Utility Menu> MenuCtrls> Update Toolbar**. You can size the Toolbar to make it smaller or larger to fit your set of abbreviations.

Avoid using ANSYS command names when naming abbreviations.

Figure 4.17 Edit Toolbar / Abbreviations Dialog Box



The order in which you define abbreviations determines the placement of the buttons on the Toolbar. Once you have defined buttons, you cannot rearrange them graphically within the GUI. (You can save the abbreviations on a file and then edit the file if you want to reorder the way the buttons appear on the Toolbar.)

Buttons that execute commands or functions from a processor other than the current one will not work. For example, if you are in PREP7 and you pick a button for a POST1 command, you will receive an "unrecognized PREP7 command..." warning.

Once you have created your own set of abbreviations in the Toolbar, you can save them on a file by choosing either **Utility Menu> MenuCtrls> Save Toolbar** or **Utility Menu> Macro> Save Abbr**, or by using the **ABBSAV** command. To restore a set of abbreviations from a file, pick either **Utility Menu> MenuCtrls> Restore Toolbar** or **Utility Menu> Macro> Restore Abbr**, or use the **ABBRES** command.

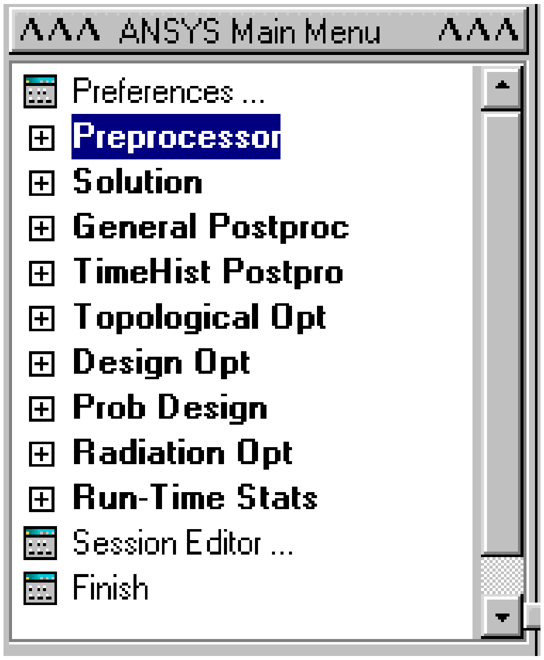
You can make additional changes to your ANSYS Toolbar by modifying the start up file. See Section 4.4.8: Creating, Modifying and Positioning Toolbars for more information.

4.4.5. The Main Menu

The Main Menu is where you begin your analysis. It contains the ANSYS analysis functions you use to create your model and perform your analysis. The Main Menu is arranged in a tree structure. This structure makes progressive submenus accessible as you proceed through the steps of the analysis. Each menu topic in the main menu either expands to show more menu options (indicated by a boxed **+**) or performs an action (indicated by an icon preceding the menu topic). Click on the boxed **+** or the topic name to expand a menu topic to reveal the subtopics (the boxed **+** will turn into a boxed **-**). Continue expanding subtopics (also indicated by a boxed **+**) until you reach the desired action. The action name is preceded by an icon used to indicate the action:

- A square for a dialog box.
- An arrow for a picking menu.

Figure 4.18 Main Menu



4.4.5.1. Using Preferences to Set Menu Content

One of the most useful customizations you can perform from the GUI is to apply filtering to your menu choices. Filtering lets you grey out, or completely hide many of the functions you determine you won't need during your analysis. You use the preferences dialog box to adjust your filtering.

When you start ANSYS, no filtering is applied (default). Menu choices for all analysis disciplines (structural, thermal, electromagnetic, ANSYS Fluid, and FLOTRAN CFD) are displayed. Menu items that are not applicable are grayed out, based on the context of your analysis (element type, solver, processor, etc.). You can remove the grayed out items by choosing the appropriate discipline from within the "Preferences for GUI Filtering" dialog box.

> *Note* — Not all menu options may be visible during your analysis. If the desired option is not available, check your preferences and element types.

For example, choosing **Thermal** suppresses structural, magnetic, and fluid element types in the "Element Types" dialog box, displacements, and potentials, etc. in the "Apply Loads" and "Delete Loads" menus, and so on. Another item for which you can control the menu filtering is the methodology used for structural analysis - h-method, p-method, or LS-DYNA Explicit. By default, ANSYS uses h-method. (See the *ANSYS Structural Analysis Guide* for discussions of these structural discipline options.)

You can also change your menu hierarchy and customize it to your needs by using the User Interface Design Language (UIDL), an ANSYS-developed GUI language. (See Chapter 6, "Customizing ANSYS and the GUI".)

4.4.5.2. Additional Usability Features

When you open a Main Menu topic, the submenus stay in place until you choose a different Main Menu topic. If you do not see a menu topic, you scroll up or down in the Main Menu area until you find the topic you are looking for, or click the boxed **-** to collapse subtopics you are no longer using.

Individual topics in the Main Menu can be completely expanded to reveal each and every subtopic. Right-mouse click context-sensitive control is available within the Main Menu to expand and collapse the individual topics. Once an individual subtopic is completely expanded, you can selectively close topics to have areas at different locations under each main topic expanded. As you do so, the amount of expansion you access is recorded and replicated the next time you enter that area.

You can set your menus to automatically collapse and expand your subtopics. Use the “Collapse Siblings” feature (found in the right-mouse click menu) to set your menu expansion preferences. When you choose collapse, the subtopics you have open automatically collapse when you choose another main topic.

When many submenus are open, navigating the main menu can become confusing, especially when individual submenus are large enough to push the heading out of the viewing area. The Preprocessor subtopic alone has more than 800 nested subheadings beneath it. You use the same right-mouse click context-sensitive control to configure the main menu for selectable contrasting color display within each nested level. You can designate any color for the menu text at each level, making the transition between levels easily detectable. This makes navigating within a subtopic, and up to higher levels much easier.

4.4.5.3. Main Menu Analysis Functions

Most of the functions in the main menu are "modal" with respect to each other; that is, you must complete one function before starting the next. For example, if you are creating keypoints on the working plane, you cannot simultaneously create lines or mesh volumes. However, you can define or edit scalar parameters in the middle of creating keypoints since parameter functions are "modeless" Utility Menu functions.

Your Main Menu functions will vary according to your product and type of analysis. The following functions are included in the default, non-filtered Main Menu:

- **Preprocessor** - Enters the preprocessor (by executing the **/PREP7** command) and contains PREP7 functions such as modeling, meshing, and loads.
- **Solution** - Enters SOLUTION (by executing the **/SOLU** command) and contains SOLUTION functions such as analysis type and options, loads, load step options, and solution execution.
- **General Postproc** - Enters the general postprocessor (by executing the **/POST1** command) and contains POST1 functions such as plotting and listing of results.
- **TimeHist Postpro** - Enters the time-history postprocessor (by executing the **/POST26** command) and loads the Time History Variable Viewer. Contains POST26 functions such as defining, listing, and plotting of variables.
- **Design Opt** - Enters the design optimizer (by executing the **/OPT** command) and contains OPT functions such as defining the optimization variables, starting the optimization run, and reviewing the resulting design sets.
- **Prob Design** - Enters the PDS processor (by executing the **/PDS** command) and contains Probabilistic Design functions.
- **Radiation Matrix** - Enters the radiation matrix generator (by executing the **/AUX12** command) and contains AUX12 functions such as defining emissivities and other settings, and writing the radiation matrix.
- **Run-Time Stats** - Enters the run-time statistics module (by executing the **/RUNST** command) and contains RUNSTAT functions such as listing statistics and providing system settings.

4.4.5.4. Additional Main Menu Utilities

The default main menu also contains the following dialog boxes that are available at any time during your analysis.

- **Session Editor** - Opens the session editor. See Section 2.8: Using the Session Editor for more information.
- **Finish** - Exits the current processor and moves you to the Begin level by executing the **FINISH** command.

4.4.6. The Graphics Window

The Graphics Window is where all ANSYS graphics displays are drawn and all graphical picking is done. It is usually the largest of the GUI windows. If you change the size of the Graphics Window, ANSYS, Inc. recommends that you maintain the 4:3 width-to-height proportion.

Graphics displays are drawn in the Graphics window when you request a plot (using either the **Plot** menu or a plot command). In addition, you will see graphics displays that are generated by immediate mode and XOR mode. (These two modes are explained next.)

4.4.6.1. Immediate Mode

An *immediate mode* plot is one that is drawn automatically when you create, move, reflect, or otherwise manipulate your model. It is only a temporary graphics display meant to give you immediate feedback on the function you just executed. As a result, an immediate mode plot has two drawbacks:

1. It will be destroyed if you bring up a menu or a dialog on top of it or if you iconify, then restore the GUI.
2. Its scaling is based on the scaling for the last plot request, so if the new entity lies "outside" the boundaries of that scaled image, it will not appear in your Graphics Window. To see the new entity, simply issue a plot request.

Numbers and symbols drawn in immediate mode (such as keypoint numbers and boundary condition symbols) have a similar drawback: they will disappear when you request a plot unless they are explicitly set to "on" via the appropriate functions under the **PlotCtrls** menu. You can turn off immediate mode using the **Immediate Display** function under **Utility Menu> PlotCtrls> Erase Options**.

If you "manually" request a plot (using the **Plot** menu or a plot command), the ANSYS program calculates the graphics scaling such that the display fills the window. This is more of a "permanent" display in that it stays in place even if it is obscured by a dialog box or a menu or if the Graphics Window is iconified and then restored. See the *ANSYS Basic Analysis Guide* for details about ANSYS graphics.

4.4.6.2. XOR Mode

The ANSYS program uses this mode when something needs to be drawn or erased quickly without destroying whatever is currently being displayed in the Graphics Window. For example, XOR mode takes effect during graphical picking to highlight or unhighlight the item being picked. It is also used to display the working plane and for rubber-banding.

The advantage of using XOR mode is that it produces an instantaneous display without affecting the existing plot on the screen. The only drawback is that drawing in the same location a second time erases the display. For example, picking the same node or keypoint a second time erases the highlight. Similarly, with the working plane display on, requesting another plot without erasing the screen erases the working plane.

4.4.6.3. Capture Image Feature

A useful feature that allows you to create "snapshots" of the Graphics Window is the Capture Image function (using the **Capture Image** button in the Standard Toolbar or **Utility Menu> PlotCtrls> Capture Image**). After an image is captured (when the "snapshot" is taken), you can save it to a file and then restore it in any ANSYS session. Captured images are useful for comparing different views, sets of results, or other significant images simultaneously on the screen.

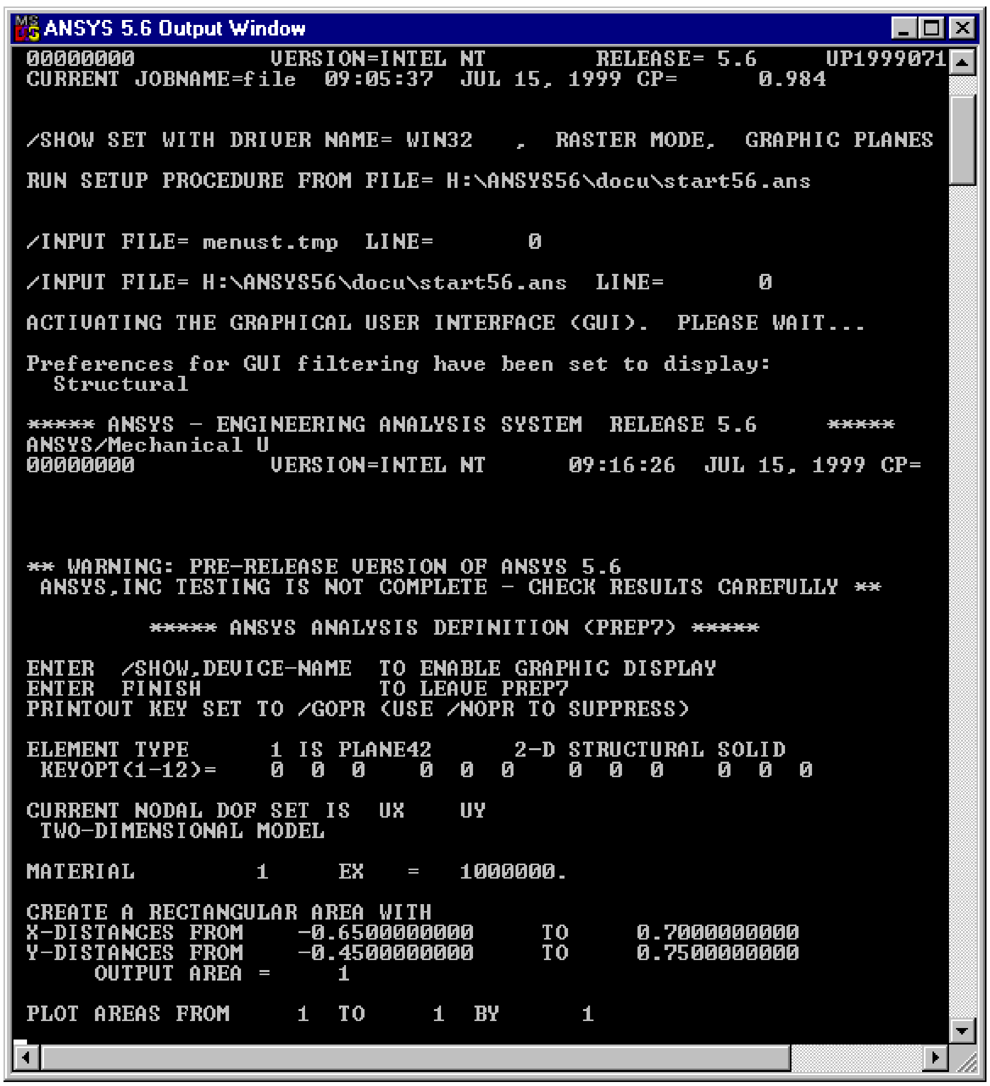
4.4.6.4. Right-mouse Button Context-sensitive Menus

You can click the right-mouse button to access many of the functions you use to adjust the Graphics Window display. The available information will vary according to the type of display and the position of your cursor in the window. Along with some of the standard Pan-Zoom-Rotate functions, you can also access many of the window control functions found in the PlotCrtrls section of the Utility Menu. Placing your cursor over the legend areas of the Graphics Window channels the context to access the legend control menus.

4.4.7. The Output Window

The Output Window receives all text output from the program - command responses, notes, warnings, errors, and any other messages. It is usually positioned behind the GUI, but you can raise it to the front when necessary.

Figure 4.19 Output Window



4.4.7.1. Using the Output Window on UNIX Systems

If you enter the ANSYS program via the launcher, the launcher creates the Output Window. It is a terminal window (such as xterm) that is also used to execute the ANSYS executable. If you enter the program by typing in the ANSYS execution command, then the window in which you typed the command becomes the Output Window.

Caution: Using your window manager to close the Output Window terminates the ANSYS session.

4.4.7.2. Sizing and Positioning the Output Window on Windows Systems

If you are running ANSYS on a Windows system, you can save the size and position of the Output Window:

1. Move the window to the desired location.
2. Size the window to the desired dimensions.
3. Select **Utility Menu> MenuCtrls> Save Menu Layout**.

4.4.8. Creating, Modifying and Positioning Toolbars

When you begin your ANSYS session, the start up routine reads a number of text files and scripts that set parameters and conditions for your ANSYS session. Many of these files can be modified to provide a more customized level of operation. The **start81.ans** file is one such file. You call up toolbars, set their position and define their content in a similar fashion.

You list the toolbars in the **tlbrlst81.ans** file. This file contains a list of the toolbars you want to activate at start up. The toolbar filenames are designated as ***.TLB** files, and each file in the list contains the specifications for the content, appearance and position of the toolbars in the ANSYS GUI. You can add additional toolbars to the GUI, (including a Pan-Zoom-Rotate functionality button bar (**ANSYSGRAPHICAL.TLB**) that is included with the program), by creating the corresponding ***.TLB** files and including them in the **tlbrlst81.ans** file. The default **tlbrlst81.ans** file loads the Standard Toolbar (see Section 4.4.2: The Standard Toolbar) by calling the file **\ANSYSSTANDARD.TLB**, and the ANSYS Toolbar (see Section 4.4.4: The ANSYS Toolbar) by calling the file **\ANSYSABBR.TLB**. These files should be placed in the same directory as your **tlbrlst81.ans** file, although the files themselves can be placed anywhere as long as the proper path string is designated and remains valid.

4.4.8.1. Creating a Toolbar File

All of the toolbar files listed above are included with ANSYS. Each of these files contains the instructions you need to create a toolbar. You can load the **ANSYSGRAPHICAL.TLB** toolbar as shipped into your **tlbrlst81.ans** file, or you can open any of the included ***.TLB** files to investigate and replicate the construction. The Standard Toolbar (**\ANSYSSTANDARD.TLB**) contains most of the program calls and definitions you would need to create your own, customized toolbar, and is listed below:

```
!===========================================================================

! Toolbars are structured in a hierarchical fashion starting with the
! name of the file which is used to read in the rest of this file. The
! filename MUST be capitalized and corresponds to the CLASS for resource
! query. The file is loaded by adding it to the tlbrlistNN.ans file.
!
!
! The entry point is the classname attribute which indicates the classnames
! used in this file.  This allows the definition of more than one toolbar
! area for toolbars in a file.
! REQUIRED
*ANSYSSTANDARD.classname: ANS_STD ANS_MAIN
!
!
```

```
! The title is displayed when the toolbar is loaded in the output
! and if expcol is 1 will be displayed on the Hide/Show bar.
! REQUIRED
*ANS_STD.title: ANSYS Standard Toolbar
!
!
! The location of the toolbar area can be n, s, e, w as position relative
! to the graphics area.  The toolbars are placed as they are read in,
! so change the order in the TLB file or tlbrlistNN.ans file.
!           n
!     -------------
!     |           |
!  w  | graphics  | e
!     |           |
!     -------------
!           s
!
! REQUIRED
*ANS_STD.location: n
!
!
! If expcol is 1 the toolbar will be displayed in a Hide/Show area so that
! the real estate can be toggled on and off.
! OPTIONAL - default is 0
*ANS_STD.expcol: 0
!
!
! The type of toolbar is either abbr or nonabbr see ANSYSABBR.TLB for
! an abbr type toolbar.
! OPTIONAL - default is nonabbr
*ANS_STD.type: nonabbr
!
! The imagedir is where the icons images will be loaded from this
! must be in Tcl form (e.g. "C:/Company ABC/Toolbar/images") or in
! a form usable by the Tcl "file join" command.
! REQUIRED
*ANS_STD.imagedir: $env(EUIDL_DIR) gui $::euidl::euidlArray(language) images
!
!
! The toolbars list allows multiple groupings within a single toolbar.
! Each toolbar will have its own frame area.
! REQUIRED
*ANS_STD.toolbars: Standard CmdPrompt UIDLPopWin AnsDynamic
!#
!# Standard Toolbar
!#
!
! The buttonlist indicates the name identifier of each control which will
! belong to this toolbar.
! REQUIRED
*ANS_STD*Standard.buttonlist: new open save sep1 panzomrot sep2 imagecap reportgen help
!
!
! The type is a standard Tcl/Tk widget type.  The least difficult is a
! simple button, others can be used but are more complicated.
! OPTIONAL - default is button
*ANS_STD*Standard.new.type: button
!
!
! The imagefile is the file to use for the icon. The image formats
! supported are: PNG (transparent), GIF, JPEG, BMP (Windows).  The
! file must be placed in the directory specified by imagedir.
! OPTIONAL/REQUIRED - if text is used then not needed
*ANS_STD*Standard.new.imagefile: new16x16.gif
!
!
! Text may be used instead of an imagefile but the imagefile overrides
! OPTIONAL/REQUIRED - if imagefile is used then not needed
*ANS_STD*Standard.new.text: New
!
!
! The command specified will be sent to the EUIDL Tcl/Tk interpreter
```

```
! which as interfaces to the ANSYS API.  Most useful would be
! ans_sendcommand to call an ANSYS command, macro, or UIDL.
! REQUIRED
*ANS_STD*Standard.new.command: catch {::euidl::databaseSetup::newAnalysis}
!
!
! The tooltip is a brief description of what this item does and is
! displayed when the mouse pointer is held over the control.
! OPTIONAL - default is no tooltip shows
*ANS_STD*Standard.new.tooltip: New Analysis
!
!
! The anscmd item is used provide a hyperlink into the help system.
! OPTIONAL - default is no hyperlinks to ANSYS help system
*ANS_STD*Standard.new.anscmd: \
 "\tSAVE" {HYPERLINK {ans_loadhelp SAVE}} "\n" {BODY1}\
 "\t/CLEAR" {HYPERLINK {ans_loadhelp /CLEAR}} "\n" {BODY1}\
 "\t/FILNAME" {HYPERLINK {ans_loadhelp /FILNAME}} "\n" {BODY1}
!
!
! What's This allows you to provide detailed information about this item.
! Use standard C formatting:
!     \\ - single slash
!     \n - newline
!     \t - tab
!     \b - backspace
! OPTIONAL - default is no "What's this?" help
*ANS_STD*Standard.new.whatsthis: \
"Saves and clears information about the existing analysis\
 and starts a new analysis.\n" {BODY1}
!
!
! The helpstatus item will show the string in the status area when the mouse
! pointer is held over the control.
! NOT IMPLEMENTED
*ANS_STD*Standard.new.helpstatus: Choose to start a new model clearing the existing one
!
!
!
*ANS_STD*Standard.imagecap.type: button
*ANS_STD*Standard.imagecap.imagefile: hardcopy16x16.gif
*ANS_STD*Standard.imagecap.command: ::euidl::graphics::captureImage::create
*ANS_STD*Standard.imagecap*tooltip: Image Capture
*ANS_STD*Standard.imagecap.anscmd: \
 "\t/UI,Format" {HYPERLINK {ans_loadhelp /UI}} "\n" {BODY1}
*ANS_STD*Standard.imagecap.whatsthis: \
"Allows the current graphic image to be captured to a printer, screen, or a\
 file.\n" {BODY1}
!
!
*ANS_STD*Standard.reportgen.type: button
*ANS_STD*Standard.reportgen.imagefile: reportgenerate16x16.gif
*ANS_STD*Standard.reportgen.command: ::euidl::report::toolbar::create
*ANS_STD*Standard.reportgen*tooltip: Report Generator
*ANS_STD*Standard.reportgen.anscmd: \
 "\tThe Report Generator" {HYPERLINK {ans_loadhelp Hlp_G_BASrstart}} "\n" {BODY1}
*ANS_STD*Standard.reportgen.whatsthis: \
"A tool for creating a report of your analysis.\n" {BODY1}
!
!
*ANS_STD*Standard.open.type: button
*ANS_STD*Standard.open.imagefile: resume16x16.gif
*ANS_STD*Standard.open.command: catch {::euidl::databaseSetup::openAnalysis}
*ANS_STD*Standard.open.tooltip: Open ANSYS File
*ANS_STD*Standard.open.helpstatus: Choose to resume the last saved model database
*ANS_STD*Standard.open.anscmd: \
 "\tRESUME" {HYPERLINK {ans_loadhelp RESUME}} "\n" {BODY1}\
 "\t/FILNAME" {HYPERLINK {ans_loadhelp /FILNAME}} "\n" {BODY1}\
 "\t/INPUT" {HYPERLINK {ans_loadhelp /INPUT}} "\n" {BODY1}
*ANS_STD*Standard.open.whatsthis: \
"Allows ANSYS database files or command input files to be read into ANSYS.\
 The file type determines the operation.  The jobname will be changed to \
```

```
 the name of the database file being resumed.\n" {BODY1}
!
*ANS_STD*Standard.undo.type: button
*ANS_STD*Standard.undo.imagefile: refresh16x16.png
*ANS_STD*Standard.undo.command: catch {ans_sendcommand Fnc_UNDO}
*ANS_STD*Standard.undo.tooltip: Session Editor
*ANS_STD*Standard.undo.helpstatus: Choose to restore the last saved model database
!
*ANS_STD*Standard.save.type: button
*ANS_STD*Standard.save.imagefile: save16x16.gif
*ANS_STD*Standard.save.command: catch {ans_sendcommand SAVE}
*ANS_STD*Standard.save.tooltip: Save Analysis
*ANS_STD*Standard.save.helpstatus: Choose to save the model to a database file
*ANS_STD*Standard.save.whatsthis: \
"Saves the current analysis to the current jobaname.db for restoration.\n" {BODY1}
*ANS_STD*Standard.save.anscmd: \
 "\tSAVE" {HYPERLINK {ans_loadhelp SAVE}} "\n" {BODY1}
!
*ANS_STD*Standard.panzomrot.type: button
*ANS_STD*Standard.panzomrot.imagefile: panzoomrotate16x16.gif
*ANS_STD*Standard.panzomrot.command: catch {ans_sendcommand )/UI,VIEW}
*ANS_STD*Standard.panzomrot.tooltip: Pan-Zoom-Rotate
*ANS_STD*Standard.panzomrot.helpstatus: Displays the Pan-Zoom-Rotate dialog
*ANS_STD*Standard.panzomrot.anscmd: \
 "\tPan, Zoom, Rotate" {HYPERLINK {ans_loadhelp Hlp_UI_PanZoom}} "\n" {BODY1}
*ANS_STD*Standard.panzomrot.whatsthis: \
"The Pan-Zoom-Rotate widget allows manipulation of the graphics screen.\n" {BODY1}
!
*ANS_STD*Standard.sep1.widgettype: separator
*ANS_STD*Standard.sep2.widgettype: separator
!
*ANS_STD*Standard.help.type: button
*ANS_STD*Standard.help.imagefile: help16x16.gif
*ANS_STD*Standard.help.command: ans_loadhelp Hlp_UI_ANSYSHelp
*ANS_STD*Standard.help.tooltip: ANSYS Help
*ANS_STD*Standard.help.anscmd: \
 "\tANSYS Help System" {HYPERLINK {ans_loadhelp Hlp_UI_ANSYSHelp}} "\n" {BODY1}
*ANS_STD*Standard.help.whatsthis: \
 "Start the ANSYS Help System.\n" {BODY1}
*ANS_STD*Standard.help.helpstatus: Choose to display help for ANSYS
!
! UIDL Raise Hidden
!
*ANS_STD*UIDLPopWin.buttonlist: uidlpop
!
*ANS_STD*UIDLPopWin.uidlpop.type: button
*ANS_STD*UIDLPopWin.uidlpop.imagefile: uidlpop16x16.png
*ANS_STD*UIDLPopWin.uidlpop.command: ::euidl::raiseWindows
*ANS_STD*UIDLPopWin.uidlpop*tooltip: Raise Hidden
*ANS_STD*UIDLPopWin.uidlpop.whatsthis: \
"Raise hidden windows to the top of the application.\n" {BODY1}
!
! Ansys Items
!
*ANS_STD*AnsDynamic.buttonlist: cntmng
*ANS_STD*AnsDynamic.availabilityCommand: ::euidl::contact::cntcMngrAvailability
!
! Contact Manager
*ANS_STD*AnsDynamic.cntmng.type: button
*ANS_STD*AnsDynamic.cntmng.imagefile: viewcontact16x16.gif
*ANS_STD*AnsDynamic.cntmng.command: ::euidl::contact::Interface
*ANS_STD*AnsDynamic.cntmng.tooltip: Contact Manager
*ANS_STD*AnsDynamic.cntmng.anscmd: \
 "\tThe Contact Manager" {HYPERLINK {ans_loadhelp Hlp_G_STR_CMAN}} "\n" {BODY1}
*ANS_STD*AnsDynamic.cntmng.whatsthis: \
"Invoke Contact Manager.\n" {BODY1}
!
! Command Prompt
!
*ANS_STD*CmdPrompt.buttonlist: cmdp
!
*ANS_STD*CmdPrompt.cmdp.type: anscmdprompt
```

```
*ANS_STD*CmdPrompt.cmdp*helpBackground: #0000AB
*ANS_STD*CmdPrompt.cmdp*helpForeground: #FFFFEE
*ANS_STD*CmdPrompt.cmdp*listHeight: 400
*ANS_STD*CmdPrompt.cmdp*width: 80
*ANS_STD*CmdPrompt.cmdp*tooltip: ANSYS Command Prompt
*ANS_STD*CmdPrompt.cmdp.whatsthis: \
"ANSYS commands may be entered into the command prompt for execution.\n" {BODY1}
!
! ANSYS Main Menu
!
*ANS_MAIN.title: ANSYS Main Menu
*ANS_MAIN.location: w
*ANS_MAIN.expcol: 1
*ANS_MAIN.type: mainmenu
```

Chapter 5: Graphical Picking

5.1. Graphical Picking

Many functions in the ANSYS program involve graphical picking - using the mouse to identify model entities and coordinate locations. There are three types of graphical picking operations:

- *Locational* picking, where you locate the coordinates of a new point
- *Retrieval* picking, where you identify existing entities for subsequent operations
- *Query* picking, where you identify points on the model for querying data

For additional information on picking, see Chapter 7, “Selecting and Components” in the *ANSYS Basic Analysis Guide*.

5.1.1. Mouse Button Assignments for Picking

Mouse button assignments during any picking operation are as follows:

- The left button picks or unpicks the entity or location closest to the mouse pointer. When you press this button, you can interrogate the entity numbers before they are picked, and thus query data. Press and drag the left mouse button to preview the item being picked or unpicked.
- The middle button (shift-right button on a two-button mouse) is the same as the Apply button on the picker. You can pick a number of items, and no operations will take effect until you hit Apply or OK. If there are no items currently picked, the middle button closes the picker.
- The right button toggles between pick and unpick mode. This button is the same as the Pick and Unpick toggle buttons on the picking menu.

5.1.2. Hot Spots

Hot spots are the locations on an entity that identify it for retrieval picking. For example, if two elements are adjacent, the element that is picked is the one whose hot spot is closest to the mouse pointer. For areas, volumes, and elements, the hot spot is at a location near the centroid. Lines have three hot spots: one in the middle and one near each end.

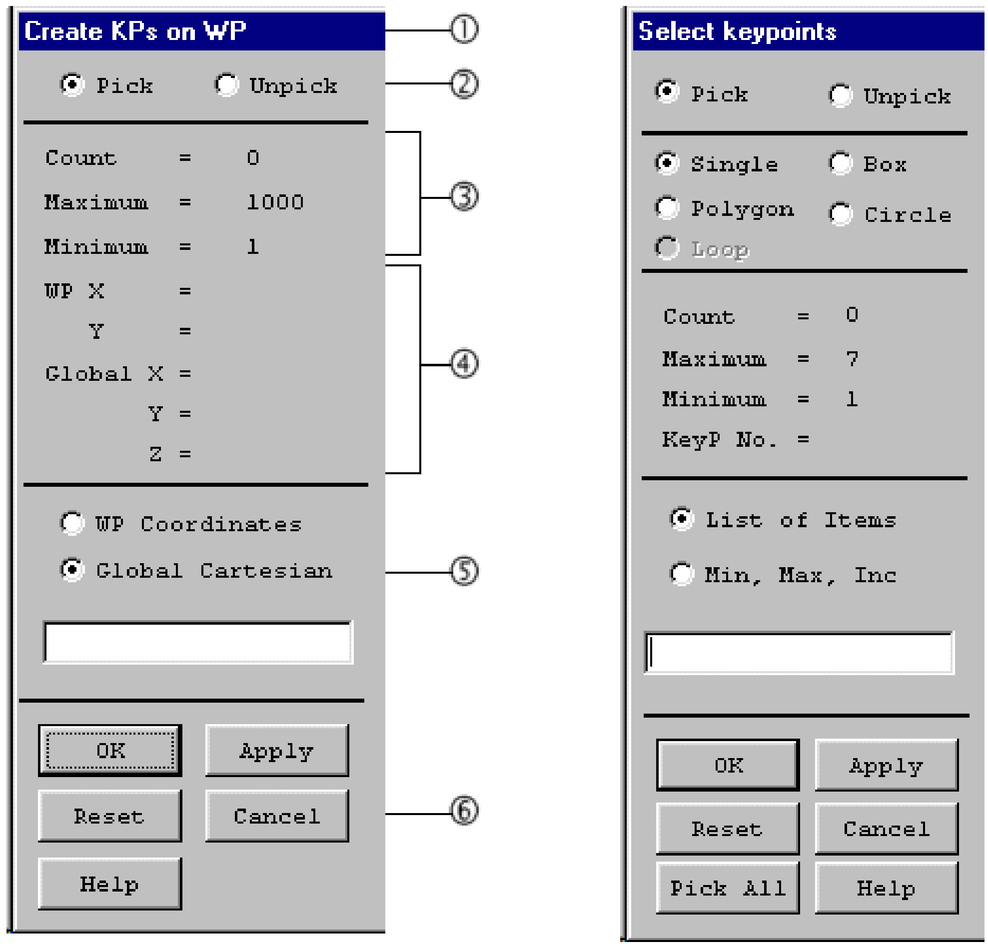
If the hot spots of two or more entities are coincident (such as for concentric circles), picking the coincident entities brings up a "Multiple Entities" dialog box (shown in Figure 5.1: “Multiple Entities Dialog Box”). Press the **Next** or **Prev** button to "cycle through" the coincident entities until the desired entity is highlighted. Then click on **OK** to pick that entity.

Figure 5.1 Multiple Entities Dialog Box

5.2. Locational and Retrieval Picking

Whenever you use graphical picking (that is, when you click on a menu topic ending with the **+** symbol), the GUI brings up a picking menu, sometimes known as the *picker*. The following figure shows the picking menus for locational and retrieval picking. In this example, creating keypoints by picking their locations on the working plane is a locational picking operation, and picking those keypoints to apply a load on them is a retrieval picking operation.

Figure 5.2 Picking Menus for Locational and Retrieval Picking



- **Function Title** [1] - Identifies the function being performed.
- **Pick Mode** [2] - Allows you to *pick* or *unpick* a location or entity. You can use either these toggle buttons or the right mouse button to switch between pick and unpick modes. The mouse pointer is shown as an up arrow for picking and a down arrow for unpicking.

For retrieval picking, you also can choose among single, box, polygon, circle, and loop mode. In single pick mode, each click on the mouse picks one entity. With box, polygon, and circle modes, you press and drag the mouse to enclose a set of entities in a box, polygon, or circle.

Loop mode is available for picking lines and areas only. With loop mode, when you pick on a line (or area), then the complete set of lines (or areas) defining a continuous loop including that line (or area) is also picked. This feature is useful when you want to identify continuous lines in order to make an area (or continuous areas to make a volume). For all modes of retrieval picking, ANSYS shows the picked entities highlighted for visual clarity.

- **Pick Status** [3] - Shows the number of items picked ("Count") and the minimum and maximum number of picks required for the function.
- **Picked Data** [4] - Shows information about the item being picked. For locational picking, the working plane and global Cartesian coordinates of the point are shown. For retrieval picking, this area shows the

entity number. You can see this data by pressing and dragging the mouse in the graphics area. This allows you to preview the information before releasing the mouse button and picking the item.

- **Keyboard Entry Options** [5] - In some cases, you may need to enter the required data by keyboard in the picker. For example, to specify a known coordinate location during locational picking, it may be easier to enter the coordinates than to use the mouse. In that case, you can choose between working plane coordinates and global Cartesian coordinates. For retrieval picking, you can choose between entering a list of entity numbers (such as 1,21,343,...) and a range of numbers (such as 1,21,2).

  Note — You must press enter to accept the selection in the text entry box before clicking **OK**.

- **Action Buttons** [6] - This area of the menu contains buttons that take action on the picked entities, as follows:

OK - Applies the picked items to execute the function and closes the picking menu.

Apply - Applies the picked items to execute the function but does not close the picking menu. You can use this button on the menu or stay in the graphics area and click the middle mouse button to apply.

Reset - Unpicks all picked entities and restores the menu and the graphics area to their state at the last Apply.

Cancel - Cancels the function and closes the picking menu.

Pick All - Picks all entities, executes the selected function, and closes the picking menu. This feature is available for retrieval picking only.

Help - Brings up help information for the function being performed.

5.3. Query Picking

You can use the picker to “Query” data on your model at any time during the analysis. Querying means that, for any picked point on the model, ANSYS will retrieve or calculate specified items from the database and display them. This information can be displayed in an output window, or it can be applied to the model as 3-D annotation. Query picking is a retrieval picking function, with the selection box items varying according to the requested data. The model and results query pickers are discussed in the following sections.

5.3.1. The Model Query Picker

The model query picker allows you to access model information by picking displayed entities. It also performs basic computations to provide simple geometric/loading information (force per unit area, angle between lines, etc.). You can use the model query picker to obtain information about the model while you are building it. This is helpful when you are building on to an existing model, or when you wish to apply forces or loads that are dependent on the model data. You access the model query picker by choosing **Utility Menu> List> Picked Entities**. The model query picker can be accessed at any time during your analysis.

5.3.1.1. Annotation

When you use the model query picker to obtain model information, ANSYS displays the information in a text window. You can also apply that information to the model as 3-D annotation by checking the “Generate 3D Anno” box on the picker. The annotation is applied with the appropriate units displayed, and arrows and leaders are attached where necessary. 3-D annotation, unlike 2-D, will attempt to retain the proper positioning on the appropriate entities, even when the model is rotated or resized.

The overall 3-D dimensions of your model are defined by a bounding box. If portions of your model's bounding box lie outside of the visible area of your graphics window (if you are zoomed in on a specific area of your model), it can affect the placement of your 3-D annotations. Zooming out will usually overcome this problem.

You modify query annotation the same way you do any other ANSYS 3-D annotation (**Utility Menu> PlotCtrls> Annotation> Create 3D Annotation**). Select "Options" to modify the typeface and move, copy or resize the annotation.

5.3.1.2. Action Buttons

- **OK** - Enters all query data and either displays it in a window, or applies it to the model. The model query picker is then closed.
- **Apply** - Enters all query data and displays it in a window, or applies it to the model. The model query picker is left open for additional operations.
- **Reset** - Clears all query data (restores the graphics display without replotting).
- **Cancel** - Clears all query data (restores the graphics display without replotting) and closes the model query picker (same as OK).
- **Pick All** - Accesses all of the selected query items on the model.
- **Help** - Brings up this screen of help information.

5.3.1.3. Tips on Using the Model Query Picker

The model query picker is accessible at any time (pre, post, solution, etc.) during your analysis. It is useful for determining model database information that would require hand calculations, or exiting a feature to determine. The computations performed by the picker are not displayed until you hit **OK** or **APPLY**.

The items selected by the picker are highlighted before they are actually selected. For complex models, or coincident viewing angles, you can hold the left mouse button down and move the pointer around the model to ensure that the proper entity is selected. You can select multiple items, with no computations being performed or annotation applied until you click OK. The commands and operations performed by the model query picker are not written to the log file.

The annotation that you generate during the preprocessing stage can be displayed at any time during your analysis. This information is not modified or updated during the course of the analysis, and can become invalid. For instance, when a shape is deformed by the applied forces, the area and volume annotations created before applying the force will not change, and the information displayed may be incorrect.

5.3.2. The Results Query Picker

Perhaps the most convenient way to review results for specific points on the model when using the General Postprocessor is by using the results query picker. Querying means that, for any picked point on the model, the ANSYS program will retrieve a specified results data value from the database and display it. You access the results query picker by choosing **Main Menu> General Postproc> Query Results**. You can query nodal, element, or subgrid (for p-elements) solution data.

When you choose this GUI path, the results query picker for element solution data appears. This menu has most of the same items as the locational and retrieval picking menus:

- **Function Title** - Identifies the information being queried; either nodal, element, or subgrid solution data (for example, element solution data).

- **Pick Mode** - Allows you to *pick* or *unpick* a location on the model to be queried. You can use either these toggle buttons or the right mouse button to switch between pick and unpick modes. The mouse pointer is shown as an up arrow for picking and a down arrow for unpicking.
- **Picked Location** - Shows the number and location (in global Cartesian coordinates) of the node, element (coordinates are of the element centroid), or subgrid point being picked for query. You can preview this information by pressing and dragging the mouse in the graphics area. The picked location is highlighted on the model as a square on the *displaced* shape.
- **Queried Data** - Shows the queried solution data for the node, element, or subgrid point at the picked location. Use the Max or Min buttons to automatically obtain the maximum or minimum value of the query data over the entire model.

5.3.2.1. Annotation

The results query picker retrieves solution data from the database and displays it in a text window. You can also apply that information to the model as 3-D annotation by selecting the “Generate 3D Anno” box on the picker. The annotation is applied with the appropriate units displayed, and arrows and leaders are attached where necessary. The overall 3-D dimensions of your model are defined by a bounding box. If portions of your model's bounding box lie outside of the visible area of your graphics window (if you are zoomed in on a specific area of your model), it can affect the placement of your 3-D annotations. Zooming out will usually overcome this problem. Unlike the model query picker, the results query picker displays the actual data on the model as you move the mouse pointer over the different entities. This function is available only from the General Postprocessor.

5.3.2.2. Action Buttons

- **OK** - Clears all query data (restores the graphics display without replotting) and closes the results query picker.
- **Apply** - Clears all query data (restores the graphics display without replotting) and brings up the Query Solution Data (either Nodal, Element, or Subgrid) dialog box again. You can use this button on the picking menu or stay in the graphics area and click the middle mouse button to achieve the same effect.
- **Reset** - Clears all query data (restores the graphics display without replotting).
- **Cancel** - Clears all query data (restores the graphics display without replotting) and closes the query picking menu (same as OK).
- **Min** - Shows the minimum value of the query data over the entire model.
- **Max** - Shows the maximum value of the query data over the entire model.
- **Help** - Brings up this screen of help information.

5.3.2.3. Tips on Using the Results Query Picker

- Query picking is most effective when used in *vector mode* (in wireframe displays), because you can read the queried data in the Graphics Window, right on top of the model. You access vector mode by choosing **Utility Menu> PlotCtrls> Device Options** (or by issuing the **/DEVICE** command).
- Press and drag the left mouse button around on the model for an easy way to see the queried data for many different points without creating excessive clutter on the screen.
- Use the **Reset** button often to clear the screen of queried data and get a "clean" model on which to continue querying.
- The **Min** and **Max** buttons are useful for quickly identifying the location and value of the minimum and maximum query data.

- The results query picker retrieves data from the database. The operations and functions performed are not written to the log file.

Note — For any picking operation, prompts are displayed in the Input Window. The prompt message indicates exactly which items are expected to be picked, and in what sequence. When in doubt about picking, read the prompt.

Chapter 6: Customizing ANSYS and the GUI

6.1. Customizing ANSYS

You can customize the ANSYS program using a set of coded (ASCII) files. Each file has the extension **.ans**. Examples are the ANSYS start-up file, **start81.ans** (discussed in Chapter 3, "Running the ANSYS Program" of this manual and in the *ANSYS Basic Analysis Guide*), and the ANSYS configuration file, **config81.ans**.

The program uses the following search paths to find the **.ans** files:

- the current directory
- the home directory
- the ANSYS APDL directory **/ansys_inc/v81/ansys/apdl**)

These paths let you customize ANSYS at either the site level or the user level.

Note to Windows Users

Windows systems may not set a home directory by default. To set a home directory in Windows, you need to set a **HOMEDRIVE** environment variable to the desired drive letter (including the colon) and a **HOMEPATH** environment variable to the desired path. For example:

```
HOMEDRIVE=C:
HOMEPATH=\Users\Mine
```

For more information on setting environment variables, contact your system administrator or consult your Windows operating system documentation.

6.1.1. The Configuration File

When it starts up, the ANSYS program searches for its optional configuration file, **config81.ans**. This file enables you to change certain system-dependent information. Table 6.1: "Configuration File Defaults and Ranges" below lists the parameters in **config81.ans** and the default, maximum, and minimum values for each parameter.

Although you can modify any of the parameter values in **config81.ans**, typically the default values are the most efficient. The **config81.ans** file should contain only the parameters for which you plan to modify values.

Table 6.1 Configuration File Defaults and Ranges

Parameter	Default	Minimum Value	Maximum Value
NUMRESLT	1000	10	unlimited
NUM_BUFR	4	1	32
SIZE_BIO	16384	1024	4194304
NUM_VPAG	512 Windows 1024 UNIX	16	8192
NUM_DPAG	131068	NUM_VPAG	131068
SIZ_VPAG	16384	4096	131072
VIRTM_MB	64 Windows 128 UNIX	40 Windows 72 UNIX	system dependent[1]

Parameter	Default	Minimum Value	Maximum Value
NUM_PROC	1	1	# of processors on your system
CONTACTS	1000	0	unlimited

1. See Platform Certification Specifics in the *ANSYS Installation and Configuration Guide for UNIX* for more information.

6.2. Splitting Files Across File Partitions

The ANSYS program, by default, creates only one (sometimes large) file for each of the various file types, which is more efficient and less complicated than multiple files. In the past, many systems had smaller partitions that were required for some of the very large ANSYS files. To accommodate this hardware limitation and provide the flexibility to store very large files across multiple disk systems, ANSYS allows you to specify split file sizes and divide files across multiple disks. For systems with file size limitations (some Linux 32-bit and Windows 32-bit with older FAT and FAT32 file systems), ANSYS automatically splits files using a default split size of 2 GB per partition.

The sparse solver, however, uses file splitting whenever files exceed 2 GB for 32-bit operating systems or 8 GB for all other systems. You cannot currently increase the split size beyond the default limits of 2 GB or 8 GB, depending on which system is used.

The split file sizes in ANSYS are defined in units of Mwords (1024*1024 4-byte integer words). The default size for most systems for all non-sparse solver files is 32768 Mwords (128 GB). On Linux 32-bit files, the default split files size is 511 Mwords (just under 2 GB). For the sparse solver files (**file.LNxx**), the maximum split file size for most systems is 2047 Mwords (just under 8 GB), but on Linux 32-bit or Windows 32-bit systems, the maximum split file size is 511 Mwords. For sparse solver files, you cannot increase these maximums. This restriction is due to integer size limitations in the Boeing solver code.

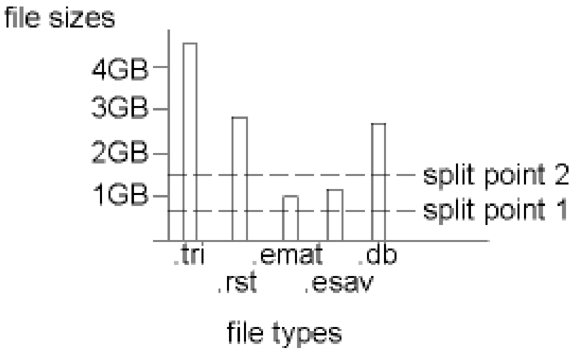
You can still use file splitting for ANSYS jobs inadvertently by leaving the FILESPLT directive in your **config81.ans** file. The default setting inside ANSYS is the recommended option in almost all cases. File splitting will almost always increase run time, particularly if different disks are used for various file partitions. The best way to improve I/O performance on modern systems is by using multiple drives which are configured in a RAID0 striped configuration.

The following guidelines will help you if you still chose to use ANSYS file splitting capabilities.

Using ANSYS' file splitting feature, you first determine a split point for your files. Then, any file types that can be split (see list below) AND are larger than that split point will be split across partitions. It is important that you choose a split point carefully. Split files are less efficient than single files, and to reuse split files, you must match the split points correctly. Therefore, you should not set your split point so low that small files would be unnecessarily split.

Figure 6.1: "Determining Split Points" illustrates one way to determine an appropriate split point. In this figure, if you chose split point 1, all of the file types would be split. If you chose split point 2, however, only three of the five file types need to be split.

Figure 6.1 Determining Split Points



You can split the following ANSYS files. See the *ANSYS Basic Analysis Guide* for file descriptions.

- **file.TRI** (or **.LN11** for the Block Lanczos solver)
- **file.FULL** (or **.REDM** for reduced modal or reduced eigenbuckling)
- **file.EMAT**
- **file.EROT**
- **file.ESAV**
- **file.MASS**
- **file.LN07**
- **file.SUB**
- **file.MODE**
- **file.LN09**
- **file.RDSP**
- **file.RFRQ**
- **file.RST**
- **file.RTH**
- **file.RFL**
- **file.RMG**
- **file.DSUB**
- **file.USUB**
- **file.DB**
- **file.LN20**
- **file.LN21**
- **file.LN22**
- **file.LN25**
- **file.PAGE**

To split files across partitions, follow these steps:

1. Issue the **RFILSZ** command in the **/RUNST** module to get an estimate of the file sizes.

2. Issue the **df** command in UNIX to determine which partitions will hold the split files. To achieve the most efficiency, create as few split files as possible.
3. Determine the least amount of space available across those partitions. This number will be the size of each split file. Adjust this number downward to be conservative.
4. Calculate the split point in terms of megawords (millions of words) by dividing the maximum size of the file to be split in bytes by 4194304 (4x1024x1024).
5. Place the **config81.ans** file in the current working directory. The file should contain the following line:

```
FILESPLT=xxxx
```

In the above line, *xxxx* is the integer value indicating the file split point in megawords. With this **config81.ans** file in place, the file will be split into a new file every increment of *xxxx* megawords. These files are all created in the current directory and are named **Jobname.*XXXnn***, where *XXX* indicates the file extension (see list, above), and *nn* is a two-digit number from 02 to 99.

To place these files in different partitions, symbolically link the name **Jobname.*XXXnn*** to a pathname on another partition with the **ln -s** command in UNIX. For example, to split a **file.TRI** and write a second file to a different partition:

```
ln -s /tmp/file.tri02  file.tri02
```

The above example puts a second **file.TRI** into the **/tmp** partition.

When you finish creating the appropriate symbolic links, you are ready to execute the ANSYS program.

You cannot transport the split files to other systems.

Caution: Unpredictable results can occur if you try to reuse a split file for substructures or restructuring and the split point has changed. ANSYS recommends discarding the split files after the run without reusing them. If you do intend to reuse the split files, the split points must match correctly. Also be aware that using split files can result in a slight performance degradation.

6.3. Customizing the GUI

The GUI allows several levels of customization, ranging from a simple change in the sizes of the GUI and the areas in it to a more complex change in the menu hierarchy and design of the dialog boxes. GUI attributes you can change include these items:

- The size of the GUI and the areas in it
- Colors and fonts
- The menus shown at GUI start-up
- The mouse and keyboard focus
- The menu hierarchy and dialog boxes.

6.3.1. Changing the GUI Layout

You can resize the ANSYS toolbar, Main Menu and Graphics Window, as well as the overall size of the GUI. To resize the areas in the GUI, drag one of the borders around the areas of the GUI while holding down the left mouse button.

To change the overall size of the GUI, position the mouse on of the corners of the GUI and drag it diagonally towards the center of the GUI while holding down the left mouse button. You can save your GUI size settings by selecting **Utility Menu> MenuCtrls> Save Menu Layout**.

Note — The "save layout" feature does not save all windows. However, it saves most of the commonly used windows, such as the picking menu, the **Pan, Zoom, Rotate** dialog box, and other dialog boxes associated with working plane, Toolbar, selecting, etc. on the Utility Menu.

6.3.2. Changing Colors and Fonts

At the system level, you can change the color, font, or both of almost any GUI component. On Windows systems, you do so via the appearance settings in the Display Properties dialog box. The fonts in the ANSYS GUI are based on your Windows font settings, as follows:

ANSYS GUI Items	Windows Item Font Equivalent
Utility Menu	Menu
Main Menu	Message Box
Tool Tip	Tool Tip

To change your fonts on Windows systems:

1. Right click on your desktop. A menu appears.
2. Select **Properties** from the menu. The Display properties dialog box appear.
3. Choose the **Appearance** tab.
4. Select the Windows item that corresponds to the ANSYS GUI item whose font you want to change.
5. Make changes to the font, size, and color and click **OK**.

On UNIX systems, you customize GUI colors and fonts by editing your .Xdefaults file. For example, to change the font size to 10 pt. Times New Roman, add the following to ~/.Xdefaults:

```
*EUIDL*Font: Times 10 normal
```

Edit your X-resource file to customize colors and fonts shown in GUI dialog boxes. If you have changed fonts in ANSYS via your .Xdefaults file in the past and it is not working, you need to change the capital F in Font to a lowercase f as in font. For example:

```
*EUIDL*font: Times 10 normal
*EUIDL*Menubutton*font Courier 10 bold
*EUIDL*MainMenu*font: Times 8 {normal italic}
```

Note — On Windows systems, set your Display settings to Small Fonts for all screen resolutions. If you use Large Fonts, your screen may not display the entire contents of large dialog boxes in ANSYS.

Within the ANSYS program, you can change the font attributes of the numbers and characters appearing on your displays by issuing the **/DEVICE**,FONT,KEY or **/DEVDISP**,FONT,KEY command. Each of these commands requires `val1` through `val6` as arguments. These arguments allow you to indicate the family name of the font that you wish to use (e.g., Courier), the weight of the font (e.g., medium), font size, and other attributes that define font selection. (See the *ANSYS Commands Reference* for detailed information about the use of `val1` through `val6` within the **/DEVICE**,FONT,KEY or **/DEVDISP**,FONT,KEY command.)

6.3.3. Changing the GUI Components Shown at Start-Up

By default, the main areas of the GUI (Utility Menu, Main Menu, Standard Toolbar, ANSYS Toolbar, Input Window, Graphics Window, and Output Window) display when you activate the GUI. The **/MSTART** command allows you to choose which GUI components are shown when the GUI is activated in addition to the main areas. (The **/MSTART** command has no equivalent GUI path.)

By including a set of **/MSTART** commands in your **start81.ans** file, you can ensure that ANSYS displays the same GUI components at the start of every session. For example, to activate the **Pan, Zoom, Rotate** dialog box at GUI start-up, the commands would be:

```
/MSTART,ZOOM,ON
```

The **/MSTART** command affects only how the GUI is *initialized*. You can always open additional GUI components once you are in the GUI.

6.3.4. Changing the Mouse and Keyboard Focus

The ANSYS program uses the concept of input *focus* to determine which window on the screen is *active*; that is, which window receives input from the mouse and the keyboard. A window may have *implicit focus* (also known as *pointer focus*), in which case moving the mouse pointer into that window makes it active. With *explicit focus*, you need to move the pointer into the window *and click the left button* for that window to be active.

The ANSYS GUI also gives automatic keyboard focus to the Input Window. This means you can enter a command without first moving the mouse into the Input Window. If you bring up a dialog box, then keyboard focus is automatically redirected to it so you can enter data without moving the mouse. This feature is sometimes called the *hot keyboard* feature.

6.3.5. Changing the Menu Hierarchy and Dialog Boxes Using UIDL

You can change the menu hierarchy to suit your analysis needs, alter the design of many of the dialog boxes and add your own macros (in the form of dialog boxes) to the menu hierarchy.

The ANSYS program reads a file called **menulist81.ans**. This file lists the names of files containing the ANSYS menu screens. The **menulist81.ans** file must be available for the ANSYS program to work.

Usually, the **menulist81.ans** file resides in the **/ansys_inc/v81/ansys/gui/en-us/UIDL** directory (UNIX systems) or the **Program Files\Ansys Inc\V81\ANSYS\gui\en-us\UIDL** subdirectory (Windows systems). However, both your current working and home directories also are searched to allow for user customization of the menu system.

To modify the ANSYS dialog boxes and main menus, you need to learn the ANSYS-developed GUI programming language, called the *User Interface Design Language (UIDL)*. For more information, refer to the *ANSYS UIDL Programmer's Guide*. (This manual is not part of the standard ANSYS documentation set; you must order it separately.)

6.3.6. Creating Dialog Boxes Using Tcl/Tk

As an alternative to using UIDL to create ANSYS dialog boxes, you can create new GUI components and dialog boxes by using the Tool Command Language and Toolkit, often referred to as Tcl/Tk.

6.4. ANSYS Neutral File Format

When you want to import a geometry file into ANSYS, you have a number of options. One is to transfer the file to a generic file format such as IGES. Many external CAD packages will export a file in this or a similar format. The IGES (International Graphics Exchange Specification) file format is a vendor-neutral standard format used to exchange geometric models between various CAD and CAE systems. ANSYS provides a number of IGES import options and specialized tools for ensuring a correct and valid model import.

You can also use one of the specifically-developed ANSYS Connection Products. These products are separately licensed ANSYS programs that allow you to import the native file format from a number of external CAD and CAE programs. Because these products are designed for each specific CAD program, the proprietary file formats are processed differently for each program. The connection products are included on your installation disk, and must be licensed separately. The documentation for them is, however, available in the ANSYS Connection Users Guide.

6.4.1. Neutral File Specification

If you are working with a proprietary format, or if you are dealing with a third party software that is not supported by ANSYS Connection, you can develop an output file that will allow you to import your model directly into ANSYS. When ANSYS imports an external drawing or part file, the information must be converted to a database. The database of a model must contain both its topology and its geometry. You use the ANSYS Neutral File format to ensure that the geometry data is properly constructed and can be read directly into the ANSYS program database. To “write” directly into the database, the file must follow certain conventions. The following sections contain the specifications for constructing this type of file.

6.4.1.1. Types of Geometric Models

Your model can fall into three geometry classifications, the wireframe, the sheet and the solid. You define each of these three types with its own specifications and requirements. ANSYS interprets these entities not by how or where they are defined in the file structure (ANSYS reads most of the data for these entities in a “free format”), but by how you manipulate and organize them. You combine various ANSYS functions and data sets to form each of these geometry classifications.

- The wireframe model consists of the multiple edges of your model, connected at their ends by common vertices. The only geometry information for the model definition is contained in the edge definition.
- The sheet model is made up of a bounded surface definition of edges. You use a sheet definition to define models where the thickness is defined a property of the sheet. ANSYS uses this structure for its finite element model, using shell elements, and defining the thickness (and other properties) with a real constant.
- The solid model is a complete and explicit definition of the 3-D model. It is made up of combinations of edges and faces, and defines both the topology (how different entities of the model are tied together) and geometry (mathematical definitions of these different entities).

How these three types of models are used and stored in different CAD/CAE programs varies. Some CAD programs require a unique entry in the database for each of the individual entities that make up a type of geometry, and they cannot be intermixed. Others make no distinction between the different entities, and the way you use them depends on how you manipulate them. ANSYS makes no distinctions. The entities can be individually defined anywhere in the database, and the definitions are manipulated using the ANSYS command structure. When a bounded face is defined in ANSYS, it can be used either as a sheet (for a two dimensional or shell analysis), or as part of a solid for a three dimensional analysis. Along these same lines, when you define an edge, you can link it to a common surface, or you can use it as part of the description of a wireframe.

6.4.1.2. The ANSYS Solid Model

ANSYS uses a Boundary Representation (B-rep) scheme to define your solid model. A B-rep model definition is based on the topological assumption that a physical object is bounded by a set of faces. These faces are regions of closed and orientable surfaces. A closed surface is one that is continuous, without breaks. An orientable surface is one that allows you to distinguish between its two sides by using a surface normal to point to the outside of the model. Each surface is bounded by edges, and each edge is bounded by vertices. Thus a B-rep solid (or body) consists of faces, edges, and vertices linked together to ensure the topological consistency of the solid. By definition, an edge must be shared by two faces , and all of the edges bounding a face must share common vertices.

Another important characteristic of a B-rep solid is its geometry. You must delineate each entity with a geometric definition. You define vertices with a set of 3-D coordinates. You define edges with curve data that includes vertices, and you define faces with surface definitions that contain edge definitions. ANSYS recognizes only NURB (Non-Uniform, Rational, B-spline) definitions for both curves and surfaces. A NURB is a parametric definition of a synthetic curve or surface. In additions, the NURB must be non-periodic. That is, the knots used for the definition must have multiplicity in the first and last knot equal to the NURB order. Any other types of geometric definitions you wish to import into an ANSYS database must be converted to a NURB format.

In addition, you must also observe the following restrictions when importing model data into ANSYS:

- Your underlying surface definition must include all of the edges. This means the underlying surface must be larger than the list of trimming edges.
- The location of your edge must agree with your surface definition. Each edge is checked to make sure that it lies on the surface within a prescribed tolerance.
- Periodic or closed curves are not allowed. You cannot have an edge that spans 360 degrees (you must break it into two 180 degree edges) and
- Periodic or closed surfaces are not allowed. You cannot have a surface that spans 360 degrees. A cylinder (for example) cannot be a single 360 degree surface (it must be broken into two 180 degree face, each having four edges). The underlying curve or surface definition must also be divided accordingly.

The ANSYS database object allows multiple bodies (or solids) to be defined. Also, the geometry of the entities of each body is stored in global 3–D space. No transformations are made or stored in the database. The following table summarizes each solid entity and its nomenclature as used in ANSYS.

Topology	ANSYS Term	Description
Object	Model	A list of bodies or volumes
Body	Volume	A list of Shells (most models will typically have only one shell)
Shell	Shell	A list of Faces
Face	Area	A list of Loops and a surface definition (only one external loop is allowed for each face)
Loop	Loop	A list of Edges
Edge	Line	Defined by a starting point, an ending point and a surface definition
Vertex	Keypoint	X, Y, Z location in Global Space

The ANSYS database also uses the concept of unique integer identifiers (ID's) to store, access and manipulate the geometry data. You can use the same number(s) for different entities, but each number must be unique within the entity. The ANSYS Neutral File uses this concept of unique ID's to guarantee valid topological relationships.

6.4.1.3. ANSYS Neutral File Functions

The four AUX15 utility functions used to transfer geometry data into the ANSYS database are listed below:

KPT
: Writes keypoints or vertices into the database.

LCURV
This function is used to write lines or edges into the database. If wireframe geometry is being defined, this function is used to define these items. This command is also used to write the information for the edges of the faces of a volume or solid entity. The keypoints used in the LCURV function refer to specific keypoint entities, individually defined by a prior KPT function.

ASURF
You use this command to write the area or face information into the database. The list of lines in each loop refer to specific line entities that you define individually with an LCURV function. This method is also used to define the sheet geometry.

VBODY
You use this command to actually define a B-rep solid. It is made up of a series of shells that contain the list of area entities. The list of areas in each shell refers to area entities defined by the ASURF functions.

Section 6.4.2: AUX15 Commands to Read Geometry Into the ANSYS database describes the actual make up of these AUX15 commands and defines specific file structure necessary to transfer geometry into the ANSYS database.

6.4.1.4. Wireframes, Individual Surfaces, and Individual Solids

When you transfer (write) the information from an external application into the ANSYS database, you must decide whether a wireframe, sheet, or a solid model is being transferred. The following points summarize the appropriate commands you will use to accomplish this:

- If you want to write only wireframe information into the database, the functions KPT and LCURV are used. No actual faces or solids are defined, but the ANSYS lines contain a curve definition. This typically will be used to mesh beam or pipe elements.
- If you want to write sheet information into the database, the KPT, LCURV, and ASURF commands are used. The ANSYS areas will contain a surface definition, but the thickness of the sheet will not be present. The ANSYS lines will contain a curve identification. You can specify a thickness by using the ANSYS real constant command R. Normally you use this format when you want to mesh shell or 2-D elements.
- If you want to write the information for a solid model into the database, you must use the KPT, LCURV, ASURF, and VBODY commands to define the geometry. The lines contain the curve definitions and the areas contain the surface definitions. The face definition also includes an expression of the direction normal to the face, so that a valid solid model is defined. This method combination of commands and topology data is used to mesh 3-D solid elements, typically 10–node tetrahedral elements.

6.4.2. AUX15 Commands to Read Geometry Into the ANSYS database

When you generate a file to be read into the ANSYS database, you must follow certain conventions. The following AUX15 commands can be interpreted by ANSYS to import your external geometry into the program. A sample Neutral File is included. Portions of this file are referenced for each command.

6.4.2.1. KPT Command

The **KPT** command defines a specific keypoint in terms of its X, Y, Z coordinates. The command format is:

LINE 1:	**KPT**,*vnum*,`X-coord, Y-coord, Z-coord`
LINE 2:	CAD ID of edge (up to 40 characters).

LINE 1 contains the command (**KPT**) and the arguments. The arguments must be separated by commas, and are defined as follows:

vnum
Unique ANSYS ID of vertex (long integer)

X-coord, *Y-coord*, *Z-coord*
The global Cartesian X, Y, and Z coordinates of the vertex.

LINE 2 contains a specific CAD identification for the Keypoint. It can be up to 40 characters, and can be made up of characters or integers. The CAD ID is optional, but if none is used, a blank line is required.

Notes

The **KPT** command explicitly defines a vertex according to the Cartesian coordinates. The keypoint is referred to in subsequent functions by the *vnum* value. This unique ANSYS ID, unlike the CAD ID, must be specified.

6.4.2.2. The LCURV Command

The **LCURV** command defines a line or edge for import into the ANSYS database. If wireframe geometry is being defined, this command is used to define these items. This command is also used to define edges belonging to faces. The command is written as a multiple line data set, with the initial command statement addressing the input into the ANSYS database, the curve definition defining the type of curve to be entered, and the subsequent line(s) defining the curve. The curve definition data is entered in a free format, that is, the data is read in sequentially, according to the values expected from the curve definition. The command format is:

LINE 1:	**LCURV**, *enum*, *crv_typ*, *start_vrtx*, *end_vrtx*, *Start_param*, *End_Param* (see format below).
LINE 2:	CAD ID of edge (up to 40 characters).
LINE 3 to n	CURVE DEFINITION (see format below)

LINE 1 contains the command name (**LCURV**) and the arguments. The arguments must be separated by commas, and are defined as follows:

enum
The unique ID of the ANSYS edge (long integer).

crv_typ
Type of curve. Only B-spline (1) curves are recognized (long integer).

start_vrtx
Vertex ID at beginning of edge, specified with a **KPT** command (integer).

end_vrtx
Vertex ID at end of edge, specified with a **KPT** command (integer).

start_param
Curve parameter defining the start of the edge (real number).

end_param
Curve parameter defining the end of the edge (real number).

LINE 2 contains a specific CAD ID for the curve or edge. It can be up to 40 characters and can be made up of characters or integers. The CAD ID is optional, but if none is used, a blank line is required.

LINE 3 AND SUBSEQUENT LINES, contain the five-integer CURVE DEFINITION STATEMENT, followed by the CURVE DEFINITION DATA. This information is read in using a free format – the order of the information determines how it is assigned, regardless of whether it is on one line or many. You can enter these values as one continuous string, separated by spaces, or as discrete values, each entered on its own line.

The first five integers make up the CURVE DEFINITION STATEMENT:

form_number order rational num_knots num_cntrl_pt

The CURVE DEFINITION STATEMENT will define the order in which the CURVE DEFINITION DATA that follows will be organized. The CURVE DEFINITION STATEMENT parameters are defined as follows:

form_number
: Acceptable form_number values are listed in the **form_number values** table for LCURV.

order
: This value is related to the order of the polynomial (Basis Function) that will define the curve.

rational
: 0 = NO, not rational.

  1 = YES, rational.

Num_knots
: The array of knots will be the first set of values in the CURVE DEFINITION DATA.

num_cntl_pnts
: The control points are the Cartesian coordinates for the keypoints along the line, 3 coordinates per point.

The lines that follow are the real numbers that make up the CURVE DEFINITION data. The following table shows the order of the data and the suggested number of items per line.

CURVE DEFINITION DATA (all values are free format)	
Array of knots	four knots per line, free format
Array of Control Points	One point per line, 3 coordinates per point
Array of weights	Only used for RATIONAL curves, four weights per line.

NOTES:

The value for crv_typ in the original **LCURV** command statement (LINE 1) will determine the format for the curve definition on the subsequent lines. A value of 1 denotes a B-spline curve, and the five integer, free format will be used to write the CURVE DEFINITION information into the ANSYS database. ONLY B-SPLINE CURVES ARE SUPPORTED. The form_number signifies the actual curve definition underlying the NURB data.

Table 6.2 form_number values for LCURV

0	B-spline
1	Line
2	Circular Arc
3	Elliptical Arc
4	Parabolic Arc
5	Hyperbolic Arc
6	Degenerate
7	Conic Section
8	Piecewise Conic Section
9	Piecewise Circular Arc

The form_number signifies the actual curve definition underlying the NURB data.

Below is a complete **LCURV** input segment, including the Command statement, the ID, the CURVE DEFINITION STATEMENT, and the CURVE DEFINITION DATA (the keypoints were defined previously).

```
lcurv,2,1,2,3,0.000000,1.000000
4
0 4 0 8 4
0.000000000000000e+000 0.000000000000000e+000 0.000000000000000e+000 0.000000000000000e+000
1.000000000000000e+000 1.000000000000000e+000 1.000000000000000e+000 1.000000000000000e+000
1.000000000000000e+000 0.000000000000000e+000 0.000000000000000e+000
1.000000000000000e+000 3.333333333333333e-001 0.000000000000000e+000
1.000000000000000e+000 6.666666666666667e-001 0.000000000000000e+000
1.000000000000000e+000 1.000000000000000e+000 0.000000000000000e+000
```

6.4.2.3. ASURF Command

The **ASURF** command defines an area or face for import into the ANSYS database. You use this command to explicitly define an individual sheet, or this face definition may belong to a B-rep solid. The list of lines in each loop refers to specific line entities that you define individually with an LCURV function. The format for an ASURF functions is:

LINE 1:	**ASURF**,*FNUM*,*surf_typ*,*num_loops*,*ncrv_max*, *start_u_parm*,*end_u_parm*,*start_v_parm*, *end_v_parm*
LINE 2:	CAD ID of face (up to 40 characters).
LINE 3 to n	LOOP DEFINITION(s)
Line n+1 to m	SURFACE Definition (See Below)

Line 1 contains the command name (**ASURF**), and the arguments. The arguments must be separated by commas and are defined as follows:

fnum
: The unique ID of the ANSYS face

srf_typ
: Type of surface. Only B-spline (1) surfaces are recognized (long integer).

num_loop
: Number of loops in this surface

ncrv_max
: Maximum number of edges in any one loop for this face.

start_u_parm
: Minimum _u_ parameter defining the surface.

end_u_parm
: Maximum _u_ parameter defining the surface.

start_v_parm
: Minimum _v_ parameter defining the surface.

end_v_parm
: Maximum _v_ parameter defining the surface.

LINE 2 - This line contains a specific CAD Identification for the face or surface. It can be up to 40 characters and can be made up of characters, or integers. The CAD ID is optional, but if none is used, a blank line is required.

LINE (3 to *n*) - These lines contain a two digit loop definition (loop type and number of edges) followed by a listing of the edges that make up the face or surface. Because they are free format, the actual number of lines used will vary.

The loop definition is in the form:

loop_typ num_edges

Array of edges

Where:

loop_typ =	Loop type = 0 signifies the external loop, the remaining loops have a value of 1
num_edges =	Number of edges in the loop
Array of edges =	List of edges using the unique ANSYS ID's you defined with the **LCURV** command. If the direction of the edge is opposite its orientating for this face, then the edge number is negative.

LINE (n+1 to *m*) - This series of lines starts with the eight-integer SURFACE DEFINITION STATEMENT, followed by the SURFACE DEFINITION DATA. This information is read in using a free format - the order of the information determines how it is assigned, regardless of whether it is on one line or many. You can enter these values as one continuous string, separated by spaces, or as discrete values, each entered on its own line.

The first eight integers make up the SURFACE DEFINITION STATEMENT:

form_number u_order v_order rational u_num_knots v_num_knots u_num_cntr_pts v_num_cntr_pts

The SURFACE DEFINITION STATEMENT will define the order in which the SURFACE DEFINITION DATA that follows will be organized. The SURFACE DEFINITION STATEMENT parameters are defined as follows:

form_number
: Acceptable form_number values are listed in **form_number values** table for ASURF.

u_order
: This value is related to the order of the polynomial in the u direction (Basis Function) that will define the surface.

v_ order
: This value is related to the order of the polynomial in the v direction (Basis Function) that will define the surface.

rational
: 0 = NO, not rational.

  1 = YES, rational.

u_num_knots
: The array of knots in the U direction will be the first set of values in the SURFACE DEFINITION DATA. Typically there are four knots per line, in a free format.

v_num_knots
: The array of knots in the U direction follow. Typically there are four knots per line, in a free format.

u_num_cntl_pnts
: The control points are the Cartesian coordinates for the keypoints along the line, 3 coordinates per point, one point per line

v_num_cntl_pnts
: The control points are the Cartesian coordinates for the keypoints along the line, 3 coordinates per point, one point per line

The lines that follow are the real numbers containing the SURFACE DEFINITION DATA. The following table shows their order, and the suggested number per line.

SURFACE DEFINITION DATA (all values are free format)	
Array of knots in U direction	Four knots per line, free format
Array of knots in V direction	Four knots per line, free format
Array of Control Points	One point per line, 3 coordinates per point
Array of weights	Only used for RATIONAL curves, four weights per line.

NOTES:

The value for surf_typ in the original **ASURF** command statement (LINE 1) will determine the format for the curve definition of the subsequent lines A values of 1 denotes a B-spline surface, and the eight-integer, free format will be used to write the surface definition information into the ANSYS database. The form_number signifies the actual surface definition underlying the NURB data. ONLY B-SPLINE SURFACES ARE SUPPORTED.

Table 6.3 form_number values for ASURF

0	B-spline
1	Plane
2	Right circular cylinder
3	Cone
4	Sphere
5	Torus
6	Surface of revolution
7	Tabulated cylinder
8	Ruled surface
9	General quadratic surfacer
10	Planar, circular capping surface
11	Planar, quadrilateral surface

An example of a complete **ASURF** input segment, including the Command Statement, the ID, The SURFACE DEFINITION STATEMENT and the SURFACE DEFINITION DATA follows (the keypoints and curves were defined previously).

```
asurf,1,1,1,4,0,1,0,1
2
0 4
-4 -3 -2 -1
0 4 4 0 8 8 4 4
0.000000000000000e+000 0.000000000000000e+000 0.000000000000000e+000 0.000000000000000e+000
1.000000000000000e+000 1.000000000000000e+000 1.000000000000000e+000 1.000000000000000e+000
0.000000000000000e+000 0.000000000000000e+000 0.000000000000000e+000 0.000000000000000e+000
1.000000000000000e+000 1.000000000000000e+000 1.000000000000000e+000 1.000000000000000e+000
0.000000000000000e+000 0.000000000000000e+000 0.000000000000000e+000
0.000000000000000e+000 3.333333333333333e-001 0.000000000000000e+000
0.000000000000000e+000 6.666666666666667e-001 0.000000000000000e+000
0.000000000000000e+000 1.000000000000000e+000 0.000000000000000e+000
3.333333333333333e-001 0.000000000000000e+000 0.000000000000000e+000
3.333333333333333e-001 3.333333333333333e-001 0.000000000000000e+000
3.333333333333333e-001 6.666666666666667e-001 0.000000000000000e+000
3.333333333333333e-001 1.000000000000000e+000 0.000000000000000e+000
6.666666666666667e-001 0.000000000000000e+000 0.000000000000000e+000
6.666666666666667e-001 3.333333333333333e-001 0.000000000000000e+000
```

```
6.666666666666667e-001 6.666666666666667e-001 0.000000000000000e+000
6.666666666666667e-001 1.000000000000000e+000 0.000000000000000e+000
1.000000000000000e+000 0.000000000000000e+000 0.000000000000000e+000
1.000000000000000e+000 3.333333333333333e-001 0.000000000000000e+000
1.000000000000000e+000 6.666666666666667e-001 0.000000000000000e+000
1.000000000000000e+000 1.000000000000000e+000 0.000000000000000e+000
```

6.4.2.4. The VBODY Command

The **VBODY** command incorporates a series of previously defined vertices, curves and surfaces to form a volume or body. The command format is:

LINE 1:	**VBODY**, *bnum*, *num_shells*, *nsrf_max* (see format below).
LINE 2:	CAD ID of body (up to 40 characters).
LINE 3 to n	SHELL DEFINITION (see format below)

LINE 1 contains the command name (**VBODY**), and the arguments. The arguments must be separated by commas and are defined as follows:

bnum
: The unique ID for the ANSYS body (long integer).

num_shells
: The number of shells in this body.

nsrf_max
: The maximum number of faces in any one shell.

LINE 2 contains a specific CAD identification for he volume or body. It can be up to 40 characters and can be made up of characters or integers. The CAD ID is optional, but if none is used, a blank line is required.

LINE 3 AND SUBSEQUENT LINES contain the two-integer BODY DEFINITION STATEMENT, followed by the BODY DEFINITION DATA. This information is read in using a free format - the order of the information determines how it is assigned, regardless of whether it is on one line or many.

The first two integers make up the BODY DEFINITION STATEMENT:

shell_typ num_faces

shell_typ
: For the shell_typ, an external shell is zero, while the internal shells have value of 1.

num_faces
: This value denotes the total number of faces in this shell.

The lines that follow make up the BODY DEFINITION DATA. This data contains an array of faces, typically with ten faces specified per line, in a free format. The list of faces uses the unique ANSYS ID's, as defined earlier with the **ASURF** command. If the normal of the face is opposite of its orientation for this shell, then the face number is negative.

6.4.3. A Sample ANSYS Neutral File Input Listing

The following listing defines a cube, based at the origin of a Cartesian coordinate system, measuring 1 x 1 x 1.

```
/com,
/com, *** ANSYS GEOMETRY NEUTRAL FILE ***
/com,
/com,     ANSYS57 Connection 57 for Pro/ENGINEER V23
```

```
/com,     Build Date: 03/01/01
/com,     Model Name: box
/com,     Units: inch
/com,
/title, Model box
/aux15
kpt,1,0.000000000000000e+000,0.000000000000000e+000,0.000000000000000e+000

kpt,2,1.000000000000000e+000,0.000000000000000e+000,0.000000000000000e+000

lcurv,1,1,1,2,0.000000,1.000000
3
0 4 0 8 4
0.000000000000000e+000 0.000000000000000e+000 0.000000000000000e+000 0.000000000000000e+000
1.000000000000000e+000 1.000000000000000e+000 1.000000000000000e+000 1.000000000000000e+000
0.000000000000000e+000 0.000000000000000e+000 0.000000000000000e+000
3.333333333333333e-001 0.000000000000000e+000 0.000000000000000e+000
6.666666666666667e-001 0.000000000000000e+000 0.000000000000000e+000
1.000000000000000e+000 0.000000000000000e+000 0.000000000000000e+000
kpt,3,1.000000000000000e+000,1.000000000000000e+000,0.000000000000000e+000

lcurv,2,1,2,3,0.000000,1.000000
4
0 4 0 8 4
0.000000000000000e+000 0.000000000000000e+000 0.000000000000000e+000 0.000000000000000e+000
1.000000000000000e+000 1.000000000000000e+000 1.000000000000000e+000 1.000000000000000e+000
1.000000000000000e+000 0.000000000000000e+000 0.000000000000000e+000
1.000000000000000e+000 3.333333333333333e-001 0.000000000000000e+000
1.000000000000000e+000 6.666666666666667e-001 0.000000000000000e+000
1.000000000000000e+000 1.000000000000000e+000 0.000000000000000e+000
kpt,4,0.000000000000000e+000,1.000000000000000e+000,0.000000000000000e+000

lcurv,3,1,3,4,0.000000,1.000000
5
0 4 0 8 4
0.000000000000000e+000 0.000000000000000e+000 0.000000000000000e+000 0.000000000000000e+000
1.000000000000000e+000 1.000000000000000e+000 1.000000000000000e+000 1.000000000000000e+000
1.000000000000000e+000 1.000000000000000e+000 0.000000000000000e+000
6.666666666666667e-001 1.000000000000000e+000 0.000000000000000e+000
3.333333333333333e-001 1.000000000000000e+000 0.000000000000000e+000
0.000000000000000e+000 1.000000000000000e+000 0.000000000000000e+000
lcurv,4,1,4,1,0.000000,1.000000
6
0 4 0 8 4
0.000000000000000e+000 0.000000000000000e+000 0.000000000000000e+000 0.000000000000000e+000
1.000000000000000e+000 1.000000000000000e+000 1.000000000000000e+000 1.000000000000000e+000
0.000000000000000e+000 1.000000000000000e+000 0.000000000000000e+000
0.000000000000000e+000 6.666666666666667e-001 0.000000000000000e+000
0.000000000000000e+000 3.333333333333333e-001 0.000000000000000e+000
0.000000000000000e+000 0.000000000000000e+000 0.000000000000000e+000
asurf,1,1,1,4,0,1,0,1
2
0 4
-4 -3 -2 -1
0 4 4 0 8 8 4 4
0.000000000000000e+000 0.000000000000000e+000 0.000000000000000e+000 0.000000000000000e+000
1.000000000000000e+000 1.000000000000000e+000 1.000000000000000e+000 1.000000000000000e+000
0.000000000000000e+000 0.000000000000000e+000 0.000000000000000e+000 0.000000000000000e+000
1.000000000000000e+000 1.000000000000000e+000 1.000000000000000e+000 1.000000000000000e+000
0.000000000000000e+000 0.000000000000000e+000 0.000000000000000e+000
0.000000000000000e+000 3.333333333333333e-001 0.000000000000000e+000
0.000000000000000e+000 6.666666666666667e-001 0.000000000000000e+000
0.000000000000000e+000 1.000000000000000e+000 0.000000000000000e+000
3.333333333333333e-001 0.000000000000000e+000 0.000000000000000e+000
3.333333333333333e-001 3.333333333333333e-001 0.000000000000000e+000
3.333333333333333e-001 6.666666666666667e-001 0.000000000000000e+000
3.333333333333333e-001 1.000000000000000e+000 0.000000000000000e+000
6.666666666666667e-001 0.000000000000000e+000 0.000000000000000e+000
6.666666666666667e-001 3.333333333333333e-001 0.000000000000000e+000
6.666666666666667e-001 6.666666666666667e-001 0.000000000000000e+000
6.666666666666667e-001 1.000000000000000e+000 0.000000000000000e+000
1.000000000000000e+000 0.000000000000000e+000 0.000000000000000e+000
```

```
1.000000000000000e+000 3.333333333333333e-001 0.000000000000000e+000
1.000000000000000e+000 6.666666666666667e-001 0.000000000000000e+000
1.000000000000000e+000 1.000000000000000e+000 0.000000000000000e+000
kpt,5,0.000000000000000e+000,0.000000000000000e+000,1.000000000000000e+000

kpt,6,1.000000000000000e+000,0.000000000000000e+000,1.000000000000000e+000

lcurv,5,1,5,6,0.000000,1.000000
8
0 4 0 8 4
0.000000000000000e+000 0.000000000000000e+000 0.000000000000000e+000 0.000000000000000e+000
1.000000000000000e+000 1.000000000000000e+000 1.000000000000000e+000 1.000000000000000e+000
0.000000000000000e+000 0.000000000000000e+000 1.000000000000000e+000
3.333333333333333e-001 0.000000000000000e+000 1.000000000000000e+000
6.666666666666667e-001 0.000000000000000e+000 1.000000000000000e+000
1.000000000000000e+000 0.000000000000000e+000 1.000000000000000e+000
kpt,7,0.000000000000000e+000,1.000000000000000e+000,1.000000000000000e+000

lcurv,6,1,7,5,0.000000,1.000000
11
0 4 0 8 4
0.000000000000000e+000 0.000000000000000e+000 0.000000000000000e+000 0.000000000000000e+000
1.000000000000000e+000 1.000000000000000e+000 1.000000000000000e+000 1.000000000000000e+000
0.000000000000000e+000 1.000000000000000e+000 1.000000000000000e+000
0.000000000000000e+000 6.666666666666667e-001 1.000000000000000e+000
0.000000000000000e+000 3.333333333333333e-001 1.000000000000000e+000
0.000000000000000e+000 0.000000000000000e+000 1.000000000000000e+000
kpt,8,1.000000000000000e+000,1.000000000000000e+000,1.000000000000000e+000

lcurv,7,1,8,7,0.000000,1.000000
10
0 4 0 8 4
0.000000000000000e+000 0.000000000000000e+000 0.000000000000000e+000 0.000000000000000e+000
1.000000000000000e+000 1.000000000000000e+000 1.000000000000000e+000 1.000000000000000e+000
1.000000000000000e+000 1.000000000000000e+000 1.000000000000000e+000
6.666666666666667e-001 1.000000000000000e+000 1.000000000000000e+000
3.333333333333333e-001 1.000000000000000e+000 1.000000000000000e+000
0.000000000000000e+000 1.000000000000000e+000 1.000000000000000e+000
lcurv,8,1,6,8,0.000000,1.000000
9
0 4 0 8 4
0.000000000000000e+000 0.000000000000000e+000 0.000000000000000e+000 0.000000000000000e+000
1.000000000000000e+000 1.000000000000000e+000 1.000000000000000e+000 1.000000000000000e+000
1.000000000000000e+000 0.000000000000000e+000 1.000000000000000e+000
1.000000000000000e+000 3.333333333333333e-001 1.000000000000000e+000
1.000000000000000e+000 6.666666666666667e-001 1.000000000000000e+000
1.000000000000000e+000 1.000000000000000e+000 1.000000000000000e+000
asurf,2,1,1,4,0,1,0,1
7
0 4
8 7 6 5
0 4 4 0 8 8 4 4
0.000000000000000e+000 0.000000000000000e+000 0.000000000000000e+000 0.000000000000000e+000
1.000000000000000e+000 1.000000000000000e+000 1.000000000000000e+000 1.000000000000000e+000
0.000000000000000e+000 0.000000000000000e+000 0.000000000000000e+000 0.000000000000000e+000
1.000000000000000e+000 1.000000000000000e+000 1.000000000000000e+000 1.000000000000000e+000
0.000000000000000e+000 0.000000000000000e+000 1.000000000000000e+000
0.000000000000000e+000 3.333333333333333e-001 1.000000000000000e+000
0.000000000000000e+000 6.666666666666667e-001 1.000000000000000e+000
0.000000000000000e+000 1.000000000000000e+000 1.000000000000000e+000
3.333333333333333e-001 0.000000000000000e+000 1.000000000000000e+000
3.333333333333333e-001 3.333333333333333e-001 1.000000000000000e+000
3.333333333333333e-001 6.666666666666667e-001 1.000000000000000e+000
3.333333333333333e-001 1.000000000000000e+000 1.000000000000000e+000
6.666666666666667e-001 0.000000000000000e+000 1.000000000000000e+000
6.666666666666667e-001 3.333333333333333e-001 1.000000000000000e+000
6.666666666666667e-001 6.666666666666667e-001 1.000000000000000e+000
6.666666666666667e-001 1.000000000000000e+000 1.000000000000000e+000
1.000000000000000e+000 0.000000000000000e+000 1.000000000000000e+000
1.000000000000000e+000 3.333333333333333e-001 1.000000000000000e+000
1.000000000000000e+000 6.666666666666667e-001 1.000000000000000e+000
1.000000000000000e+000 1.000000000000000e+000 1.000000000000000e+000
```

```
lcurv,9,1,1,5,0.000000,1.000000
13
0 4 0 8 4
0.000000000000000e+000 0.000000000000000e+000 0.000000000000000e+000 0.000000000000000e+000
1.000000000000000e+000 1.000000000000000e+000 1.000000000000000e+000 1.000000000000000e+000
0.000000000000000e+000 0.000000000000000e+000 0.000000000000000e+000
0.000000000000000e+000 0.000000000000000e+000 3.333333333333333e-001
0.000000000000000e+000 0.000000000000000e+000 6.666666666666667e-001
0.000000000000000e+000 0.000000000000000e+000 1.000000000000000e+000
lcurv,10,1,2,6,0.000000,1.000000
15
0 4 0 8 4
0.000000000000000e+000 0.000000000000000e+000 0.000000000000000e+000 0.000000000000000e+000
1.000000000000000e+000 1.000000000000000e+000 1.000000000000000e+000 1.000000000000000e+000
1.000000000000000e+000 0.000000000000000e+000 0.000000000000000e+000
1.000000000000000e+000 0.000000000000000e+000 3.333333333333333e-001
1.000000000000000e+000 0.000000000000000e+000 6.666666666666667e-001
1.000000000000000e+000 0.000000000000000e+000 1.000000000000000e+000
asurf,3,1,1,4,0,1,0,1
12
0 4
10 -5 -9 1
0 4 4 0 8 8 4 4
0.000000000000000e+000 0.000000000000000e+000 0.000000000000000e+000 0.000000000000000e+000
1.000000000000000e+000 1.000000000000000e+000 1.000000000000000e+000 1.000000000000000e+000
0.000000000000000e+000 0.000000000000000e+000 0.000000000000000e+000 0.000000000000000e+000
1.000000000000000e+000 1.000000000000000e+000 1.000000000000000e+000 1.000000000000000e+000
0.000000000000000e+000 0.000000000000000e+000 0.000000000000000e+000
0.000000000000000e+000 0.000000000000000e+000 3.333333333333333e-001
0.000000000000000e+000 0.000000000000000e+000 6.666666666666667e-001
0.000000000000000e+000 0.000000000000000e+000 1.000000000000000e+000
3.333333333333333e-001 0.000000000000000e+000 0.000000000000000e+000
3.333333333333333e-001 0.000000000000000e+000 3.333333333333333e-001
3.333333333333333e-001 0.000000000000000e+000 6.666666666666667e-001
3.333333333333333e-001 0.000000000000000e+000 1.000000000000000e+000
6.666666666666667e-001 0.000000000000000e+000 0.000000000000000e+000
6.666666666666667e-001 0.000000000000000e+000 3.333333333333333e-001
6.666666666666667e-001 0.000000000000000e+000 6.666666666666667e-001
6.666666666666667e-001 0.000000000000000e+000 1.000000000000000e+000
1.000000000000000e+000 0.000000000000000e+000 0.000000000000000e+000
1.000000000000000e+000 0.000000000000000e+000 3.333333333333333e-001
1.000000000000000e+000 0.000000000000000e+000 6.666666666666667e-001
1.000000000000000e+000 0.000000000000000e+000 1.000000000000000e+000
lcurv,11,1,3,8,0.000000,1.000000
17
0 4 0 8 4
0.000000000000000e+000 0.000000000000000e+000 0.000000000000000e+000 0.000000000000000e+000
1.000000000000000e+000 1.000000000000000e+000 1.000000000000000e+000 1.000000000000000e+000
1.000000000000000e+000 1.000000000000000e+000 0.000000000000000e+000
1.000000000000000e+000 1.000000000000000e+000 3.333333333333333e-001
1.000000000000000e+000 1.000000000000000e+000 6.666666666666667e-001
1.000000000000000e+000 1.000000000000000e+000 1.000000000000000e+000
asurf,4,1,1,4,0,1,0,1
14
0 4
11 -8 -10 2
0 4 4 0 8 8 4 4
0.000000000000000e+000 0.000000000000000e+000 0.000000000000000e+000 0.000000000000000e+000
1.000000000000000e+000 1.000000000000000e+000 1.000000000000000e+000 1.000000000000000e+000
0.000000000000000e+000 0.000000000000000e+000 0.000000000000000e+000 0.000000000000000e+000
1.000000000000000e+000 1.000000000000000e+000 1.000000000000000e+000 1.000000000000000e+000
1.000000000000000e+000 0.000000000000000e+000 0.000000000000000e+000
1.000000000000000e+000 0.000000000000000e+000 3.333333333333333e-001
1.000000000000000e+000 0.000000000000000e+000 6.666666666666667e-001
1.000000000000000e+000 0.000000000000000e+000 1.000000000000000e+000
1.000000000000000e+000 3.333333333333333e-001 0.000000000000000e+000
1.000000000000000e+000 3.333333333333333e-001 3.333333333333333e-001
1.000000000000000e+000 3.333333333333333e-001 6.666666666666667e-001
1.000000000000000e+000 3.333333333333333e-001 1.000000000000000e+000
1.000000000000000e+000 6.666666666666667e-001 0.000000000000000e+000
1.000000000000000e+000 6.666666666666667e-001 3.333333333333333e-001
1.000000000000000e+000 6.666666666666667e-001 6.666666666666667e-001
```

```
1.000000000000000e+000 6.666666666666667e-001 1.000000000000000e+000
1.000000000000000e+000 1.000000000000000e+000 0.000000000000000e+000
1.000000000000000e+000 1.000000000000000e+000 3.333333333333333e-001
1.000000000000000e+000 1.000000000000000e+000 6.666666666666667e-001
1.000000000000000e+000 1.000000000000000e+000 1.000000000000000e+000
lcurv,12,1,4,7,0.000000,1.000000
19
0 4 0 8 4
0.000000000000000e+000 0.000000000000000e+000 0.000000000000000e+000 0.000000000000000e+000
1.000000000000000e+000 1.000000000000000e+000 1.000000000000000e+000 1.000000000000000e+000
0.000000000000000e+000 1.000000000000000e+000 0.000000000000000e+000
0.000000000000000e+000 1.000000000000000e+000 3.333333333333333e-001
0.000000000000000e+000 1.000000000000000e+000 6.666666666666667e-001
0.000000000000000e+000 1.000000000000000e+000 1.000000000000000e+000
asurf,5,1,1,4,0,1,0,1
16
0 4
12 -7 -11 3
0 4 4 0 8 8 4 4
0.000000000000000e+000 0.000000000000000e+000 0.000000000000000e+000 0.000000000000000e+000
1.000000000000000e+000 1.000000000000000e+000 1.000000000000000e+000 1.000000000000000e+000
0.000000000000000e+000 0.000000000000000e+000 0.000000000000000e+000 0.000000000000000e+000
1.000000000000000e+000 1.000000000000000e+000 1.000000000000000e+000 1.000000000000000e+000
1.000000000000000e+000 1.000000000000000e+000 0.000000000000000e+000
1.000000000000000e+000 1.000000000000000e+000 3.333333333333333e-001
1.000000000000000e+000 1.000000000000000e+000 6.666666666666667e-001
1.000000000000000e+000 1.000000000000000e+000 1.000000000000000e+000
6.666666666666667e-001 1.000000000000000e+000 0.000000000000000e+000
6.666666666666667e-001 1.000000000000000e+000 3.333333333333333e-001
6.666666666666667e-001 1.000000000000000e+000 6.666666666666667e-001
6.666666666666667e-001 1.000000000000000e+000 1.000000000000000e+000
3.333333333333333e-001 1.000000000000000e+000 0.000000000000000e+000
3.333333333333333e-001 1.000000000000000e+000 3.333333333333333e-001
3.333333333333333e-001 1.000000000000000e+000 6.666666666666667e-001
3.333333333333333e-001 1.000000000000000e+000 1.000000000000000e+000
0.000000000000000e+000 1.000000000000000e+000 0.000000000000000e+000
0.000000000000000e+000 1.000000000000000e+000 3.333333333333333e-001
0.000000000000000e+000 1.000000000000000e+000 6.666666666666667e-001
0.000000000000000e+000 1.000000000000000e+000 1.000000000000000e+000
asurf,6,1,1,4,0,1,0,1
18
0 4
9 -6 -12 4
0 4 4 0 8 8 4 4
0.000000000000000e+000 0.000000000000000e+000 0.000000000000000e+000 0.000000000000000e+000
1.000000000000000e+000 1.000000000000000e+000 1.000000000000000e+000 1.000000000000000e+000
0.000000000000000e+000 0.000000000000000e+000 0.000000000000000e+000 0.000000000000000e+000
1.000000000000000e+000 1.000000000000000e+000 1.000000000000000e+000 1.000000000000000e+000
0.000000000000000e+000 1.000000000000000e+000 0.000000000000000e+000
0.000000000000000e+000 1.000000000000000e+000 3.333333333333333e-001
0.000000000000000e+000 1.000000000000000e+000 6.666666666666667e-001
0.000000000000000e+000 1.000000000000000e+000 1.000000000000000e+000
0.000000000000000e+000 6.666666666666667e-001 0.000000000000000e+000
0.000000000000000e+000 6.666666666666667e-001 3.333333333333333e-001
0.000000000000000e+000 6.666666666666667e-001 6.666666666666667e-001
0.000000000000000e+000 6.666666666666667e-001 1.000000000000000e+000
0.000000000000000e+000 3.333333333333333e-001 0.000000000000000e+000
0.000000000000000e+000 3.333333333333333e-001 3.333333333333333e-001
0.000000000000000e+000 3.333333333333333e-001 6.666666666666667e-001
0.000000000000000e+000 3.333333333333333e-001 1.000000000000000e+000
0.000000000000000e+000 0.000000000000000e+000 0.000000000000000e+000
0.000000000000000e+000 0.000000000000000e+000 3.333333333333333e-001
0.000000000000000e+000 0.000000000000000e+000 6.666666666666667e-001
0.000000000000000e+000 0.000000000000000e+000 1.000000000000000e+000
vbody,1,6
cube
1 -2 -3 -4 -5 -6
```

Chapter 7: Using the Online Help and Manuals

7.1. Using the Help System

The online help system gives you information on virtually any component in the Graphical User Interface (GUI) and any ANSYS command or concept. It includes the usual features such as a cascading table of contents, index, and full text search capability. The online help is Oracle's Oracle Help for Java.

You can access the online help within the GUI in three ways:

1. Choose **Utility Menu> Help> Help Topics** and navigating to the desired topic via the table of contents and index.
2. Press the **Help** button within dialog boxes.
3. Enter the **HELP** command directly in the Input Window. When using the **HELP** command, ANSYS will dynamically complete the command. For example, to get help on the **PLNSOL** command, you could type `Help,plns`.

Note — When help is minimized and you issue another help call from ANSYS, the help window is not automatically restored to its previous size. You must do this manually.

You can also access the help system when in non-GUI mode. You can use either of the following options:

- Type **/MENU**,ON to activate the GUI and then use one of the above methods.
- Type **HELP**,*command*, as described above. This command activates the ANSYS help system without entering GUI-mode.

You can also access the help in a standalone format, without running ANSYS. Use the launcher on UNIX, the **Start** menu on Windows, or the **ANSHELP***nn* command (where *nn* is the current ANSYS release number).

7.2. Using Hypertext Links

As you scan a page of text in the navigation window, you will notice certain words or phrases are underlined and appear in a different color. These items are *hypertext links*. A hypertext link is a text navigation tool that, when clicked, takes you directly to information on that item. Typical items which appear as hypertext links are command names, element types, and manual section references.

For example, a page in the *ANSYS Thermal Analysis Guide* manual may contain several commands pertaining to steady-state thermal analysis. Each command is displayed in a color, such as dark blue. Click one of these commands, and the corresponding command description will replace the information currently in the document window. That command description will most likely also contain hypertext links that access further related information. If you return to the page where you selected the hypertext link, you will see that the color has changed. A different color indicates that you have accessed that link.

7.3. Locating Topics Via Word Search

You can locate a topic anywhere in the documentation via typical full-text search facilities, accessible within the online documentation itself.

7.4. Revisiting Previously-Viewed Topics

As you travel through various sections of the manuals, you may wish to revisit a piece of text that you scanned earlier. Use the history option or the **Back** button on your browser to review previously viewed topics.

7.5. Printing the Window Contents

You can print any currently open html help topic by using the print option on your browser. You can print only the currently open file in UNIX; however, you can print any part of the table of contents "tree" in Windows.

7.6. Using 'What's This' Help

Some ANSYS dialog boxes also feature “What's This” help. “What's This” help provides a brief description of a dialog box or an item on a dialog box. Access “What's This” help by positioning the mouse over the item of interest on the dialog box and clicking the right mouse button one time. A small context menu appears next to the mouse cursor or arrow pointer. Click on the “What's This” option and the information appears in a pop-up window on your screen. If “What's This” help is not available, the pop-up window will state that help is not available for this item. Close the “What's This” help window by clicking anywhere in the application.

7.7. Customizing ANSYS Help

ANSYS supports two separate levels of customized help.

Corporate Help
: This is help that is useful to all ANSYS users in an organization. Help files must be added to the ANSYS help directory, which could require administrative privileges on UNIX machines. On Windows-based systems, the help directory is **Program Files\Ansys Inc\V81\CommonFiles\help\en-us**. On UNIX, it's **/ansys_inc/v81/commonfiles/help/en-us**.

User Help
: This is help tailored for an individual user. These help files can reside on the user's local machine.

In neither case can you add topics directly to ANSYS help. Your help topics must be maintained as separate files. However, you can interconnect your topics as context-sensitive help accessible from custom ANSYS GUI elements or the ANSYS command line.

ANSYS can call standard HTML files (Windows and UNIX) or compiled help files (Windows only). In either case, ANSYS finds the appropriate help file through a look-up table that matches a context sensitive help string to the URL for the help file. The process of customizing help consists of:

1. Adding the appropriate help calls to whatever new or customized GUI elements you may need. This step is optional, and only applies if you are also customizing the interface. You can still access your help file by issuing the **HELP** command from the ANSYS command line.
2. Creating documentation or help topic files as HTML files.
3. Adding URLs and the help calling strings to the appropriate look-up table. File names will be resolved relative to the location of the index files.

Note — For Windows systems, a number of commercially available tools can create a compiled HTML Help file. You can also download a free tool, called the HTMLHelp Workshop, from the developer area at the Microsoft website.

Using the above as a guide, you can connect your custom help by adding look-up records to the appropriate index files.

ANSYS, Inc. supports only the functionality as described in the help system as shipped on the distribution media. Any customizations you make to the help system, including content, are solely your responsibility. We recommend that you limit your help customizations to adding additional topics.

ANSYS includes sample HTML and index files on the media, called **sample.html** and **sample.hlp** respectively. The examples in this section match those files. However, the examples in this section show HTML files called directly, and not the use of compiled help (**.chm**) files. The **sample.hlp** includes lines demonstrating **.chm** help files for your reference.

7.7.1. Adding Help Calls to GUI Objects

In general, a help string is connected to a button in a UIDL or Tcl dialog box. The string contents are completely arbitrary; however, the length must not exceed 16 characters and should be made up of letters, numbers, and additional characters such as underbars, dashes, etc. Spaces are not allowed. A typical UIDL help call (called a :H line) looks like this:

```
:H Hlp_sample
```

See the *ANSYS UIDL Programmer's Guide* for more information on the :H line.

In Tcl, the same help call issuing an action for the help button would look like this:

```
$dialogName.parent.btnHelp configure \
      -command {ans_loadhelp Hlp_sample}
```

If you want to be able to also call your HTML file from the ANSYS command line (using the **HELP**,`helpstring` command), simply use a help string that is all uppercase letters. In fact, you can use this technique to access help files without having them connected to any GUI objects.

7.7.2. Creating HTML Files

You can use virtually any tool that creates ”clean" HTML files to build your HTML files. If you wish to use compiled HTML files (**.chm**) on Windows systems, you can use any of the commercial help authoring tools or the freeware HTMLHelp Workshop which can be downloaded from the Microsoft web site.

While the limitations and capabilities of the Microsoft Internet Explorer browser and the related HTMLHelp Viewer are well known and documented on the Microsoft web site, the capabilities of the ANSYS UNIX help browser is not general knowledge. From a help authoring perspective, the display engine in UNIX is OracleHelp, and specific information can be found on the Oracle web site. In general, the OracleHelp browser supports HTML 4.0, unicode characters, and both CSS 1 and CSS 2.

Where you place your custom help files depends on who you wish to be able to access those files. If the help files should be accessible by all ANSYS users on at your site, the custom help files should be placed in the following locations:

- For Windows, place them in either **Program Files\Ansys Inc\V81\CommonFiles\help\en-us** or a subdirectory of that directory.
- For UNIX, place them in the **/ansys_inc/v81/commonfiles/help/en-us** directory. Placing files here will require administrative authority on that directory.

If you are creating help that is accessible to only an individual user, place the help anywhere on the machine used to run ANSYS. For example, if the user is running a Windows version of ANSYS, place the help files in a directory on that machine. However, if the user is running ANSYS on a remote machine, place the files somewhere on the remote machine.

7.7.3. Updating the Look-Up Table

Regardless of whether you're building online documentation for Windows or UNIX, as far as a help author is concerned ANSYS calls help in largely the same way. Both platforms use a lookup table (called an "index file") to match up a help string with an ANSYS help topic. The help string can be issued by a GUI object, such as a dialog box built in either Tcl or UIDL, or it can be issued by a **help**,`helpstring` command. The help topic can be identified with either

- a URL for an HTML file (all platforms)
- a specialized URL pointing to an HTML file within a **.chm** compiled HTML file (Windows HTML Help only)

ANSYS use three look-up files (index files). Two of the index files reside in the following location in the ANSYS installation tree:

/ansys_inc/v81/commonfiles/help/en-us.

- The first index file is supplied by ANSYS and maps the help calls to the corresponding ANSYS-supplied help topics and is called **ansys_Index.hlp** on UNIX and **win_ansys_Index.hlp** on Windows systems. **Do not modify this file!**
- The second file controls access to “corporate help" or custom help that will be accessible by all ANSYS users in an organization. It is provided as a template which you can populate. The file maps any customized help calls to their corresponding user-supplied HTML help or standard HTML files. This file is called **corp_Index.hlp**.
- The third file, which you must create, is used to point to custom help files intended for an individual users use. The file is called **user_Index.hlp**. This file is used in conjunction with an environment variable you must set called **USERHELP**, the value of which should be the complete path to the directory that contains the user help index file. To create this file, simply copy and rename **corp_Index.hlp** file. For convenience sake, you should place your custom help files in the same directory.

  Note — Do not set the **USERHELP** environment variable until you've placed the user help index file in the directory to which **USERHELP** points. If you do, ANSYS will disable the help system during its current run when it can't find the appropriate index file.

A help string, such as *Hlp_sample*, must reside in an index file and map to the appropriate topic. In this case, the line could look something like this:

```
Hlp_sample sample.html
```

- The first field contains the ANSYS help string (*Hlp_sample*).
- The second field contains the “target” topic within the help system. For UNIX, this is simply the name of the HTML file. For a Windows system, the second field can be either the name of an HTML file or the location of a topic in a compiled help file.

  For example, the target topic for an HTML file for either Windows or UNIX platforms would look like this:

  ```
  sample.html
  ```

  The target topic for a compiled help file on Windows platforms would look like this:

  ```
  sample.chm::/sample.html
  ```

To connect the UIDL help call to the help file, include a line in the index file similar to:

```
Hlp_sample  sample.html
```

The index file for the examples shown here would look like this:

```
4 13 24
Hlp_sample Hlp_sample.html              !Opens a single HTML file from the GUI
Hlp_samplechm sample.chm::/sample.html  !Opens a .chm file from the GUI (Windows only)
SAMPLE Hlp_sample.html                  !Opens a single HTML file from a HELP
command
SAMPLECHM sample.chm::/sample.html      !Opens a .chm file from a HELP command (Windows only)
```

Note that this file includes comments for each line; the actual file cannot include comments.

The GUI help call for the single HTML file for this example would be:

```
:H Hlp_sample
```

When updating the corporate or user index file, note that the first record of the index file (header record) contains three numeric fields:

- The first number represents the number of lines in the file excluding the header record.
- The second number is the length of the longest string in field one for the file.
- The third number is the length of the longest string in field two for the file.

You must update this record before the corporate index file will work properly with ANSYS. When creating customized help, follow these additional guidelines:

- Each instance of the string in field one must be unique within the file.
- Each record must map to one and only one HTML file.
- More than one record can map to the same HTML file.
- The records in the index file must be sorted on the contents of field one.

The index files are searched in the following order of precedence, and ANSYS uses the first record that matches in its search.

1. **ansys_Index.hlp** (or **win_ansys_Index.hlp** on Windows)
2. **corp_Index.hlp**
3. **user_index.hlp**

Chapter 8: Using the ANSYS Command Log

8.1. Logs ANSYS Creates

The ANSYS program records every command it executes, whether typed in directly or executed by a function in the Graphical User Interface (GUI), in two places: the session log file and the internal database command log.

- The session log file is a text file which is saved in your working directory.
- The database command log is saved in the ANSYS database (in memory). You can copy this log to a file at any time by choosing **Utility Menu> File> Write DB Log File** or by issuing the **LGWRITE** command.

Both files are command logs that can be used as input to the ANSYS program. (The general term *command log file* indicates both the session log file and the database command log produced by **LGWRITE**.)

8.2. Using the Session Log File

Every ANSYS session produces a session log named **Jobname.LOG**. (Jobname is determined by the jobname defined at ANSYS entry - see Chapter 3, “Running the ANSYS Program”. The default jobname is **FILE** or **file**, depending on the operating system.

The program opens the log file when you first enter the program, and closes it when you exit the program.

The session log file provides a complete record of your ANSYS session (in terms of commands) and is quite valuable as a means of recovering from a system crash or catastrophic user mistake. By reading in a renamed copy of your log file (or by submitting it as a batch file), you can re-execute every command on your log file, recreating your database exactly as it existed previously. (For more information, see Section 8.3: Using the Database Command Log.)

Your log file is also useful as a debugging tool that can help to reveal any mistakes you might have made in an ANSYS session. Should you require help from your ASD in debugging an ANSYS session, he or she will almost certainly ask to see a copy of your log file.

Each new ANSYS session appends commands to the existing file **Jobname.LOG**. (That is, the log file is not overwritten during a new ANSYS session, but added to.) A "time stamp," consisting of the current date and time, is included so that you can identify the start of each session. Use **/FILNAME**,1 to start a new log file for the session.

You can list your entire log file during an interactive run by picking **Utility Menu> List> Files> Log File**. Because this file is in ASCII format, you can view and edit it readily using an external text editor.

8.3. Using the Database Command Log

ANSYS captures commands generated (or typed in) during an ANSYS session not only in the log file but also in memory. This in-memory version of the command history is called the internal database log. When you save the database, the program saves this command log in the database file (**Jobname.DB**) along with the other database information.

Use either **Utility Menu> File> Write DB Log File** or the **LGWRITE** command to write the database command log to a named ASCII file. You can then edit this file, make desired changes, and use the file as command input to the program. This capability is especially useful if you want to use the command history that was created during an interactive session, but have somehow lost or corrupted the session log file (**Jobname.LOG**) that was associated with your database.

If you create your database in multiple sessions by saving and resuming the database file, the ANSYS program keeps the database log continuous by appending each new command that is processed. Therefore, the internal database log is not fragmented; it will represent the complete database.

Note — The **RESUME** command is not written to the database log.

8.4. Using a Command Log File as Input

The procedure for re-executing the commands contained in a **Jobname.LOG** file or in the database log consists of three main steps:

1. Establish the command log file.
2. Edit the command log file as desired.
3. Read the edited file into your ANSYS session.

Step 1: Establish the Command Log File

The method to do this depends on whether you use the session log file or the database log.

Session Log File

To establish a command log file from the session log file, (**Jobname.LOG**) perform these steps:

1. Rename or copy the session log file to a different name. You can do this at the system level or from within the program.
2. List the log file by choosing **Utility Menu> List> Files> Log File**.
3. Choose **File> Save as** from the Log File window.

Database Command Log

To establish a command log file from the database log, pick **Utility Menu> File> Write DB Log File**, or use the **LGWRITE** command. You can specify a file name or use the default name, **Jobname.LGW**. You also have the option (with the `Kedit` field) to write *all* commands (default), essential commands only (`Kedit` = REMOVE), or essential commands with nonessential commands commented out (`Kedit` = COMMENT).

Keep these additional points in mind when using the database command log:

- ANSYS does not store commands that are read from macros in the database command log. In such cases, you cannot use the LGW file to recreate an exact copy of the database unless all of your macros still exist and are accessible.
- Files that are read with the **/INPUT** command will be recorded in the database log (and on **Jobname.LOG**) *only* if the `LOG` field on **/INPUT** is set to 1.
- Some "nonessential" commands that are filtered out by the `Kedit` option on **LGWRITE** may be required for subsequent ***GET** operations (for example, **EXTREM** and **PLNSOL**). Therefore, use `Kedit` = COMMENT (not `Kedit` = REMOVE) to write nonessential commands as comment lines. Using a system editor, you can then "uncomment" the commands required by ***GET** to ensure exact recreation of the database.
- When you use the **RESUME** command to load a previously saved database, the program clears the database log and replaces it with the database log that is stored on the resumed database. (The **RESUME** command itself is not written to the database command log.)

- The **/CLEAR** command (**Utility Menu> File> Clear & Start New**) clears both the ANSYS database and the database log. The ANSYS program stores any commands processed after **/CLEAR** in the new database that is being created. (The **/CLEAR** command is not written to the database command log.)
- When you run ANSYS in batch mode, the program copies all commands in the input file to the database log before any of these commands execute. If the batch input contains one or more **/CLEAR** commands, the database log will not match the contents of the database that is produced.

Step 2: Edit the Command Log File

Sometimes, you will need to edit your command log file before using it as program input. As you edit your log file, you may want to add comments or indentation to improve its readability. You can add comments to your log file by using comment commands (such as **/COM** or **C*****) or by using the comment character (!).

Comments contained in a comment command will appear in both the input and output listings. Comments that follow a ! will appear only in the input listing. You can indent commands to visually group the commands for clarity.

> *Note* — Certain commands, such as **FLST** and **FITEM**, are generated by the GUI and are not intended to be typed in directly. You should avoid editing such commands on the log file. Any data change within the **FITEM** command, for example, could render the data to be invalid, and could cause unpredictable results.

Step 3: Read in the Edited Log File

In an interactive session, pick **Utility Menu> File> Read Input from** or issue the **/INPUT** command to read in the edited command log file. In batch mode, you can use the edited command log file as your batch input file.

Index