5. ANSYS Program Interaction

At the time of this writing ANSYS has been evolving for nearly thirty years into the wealth of capability that it provides. In the past the primary way to interact with the ANSYS program was via commands; secondarily using the User Programmable Features. The direction in the future will be to use C API calls or Tcl/Tk to interact with ANSYS. This chapter will focus on how to interact with ANSYS using custom Tcl/Tk ANSYS commands.

5.1 Invoking Tcl/Tk from ANSYS

The external command feature of ANSYS handles the invocation of Tcl/Tk. There are three default ways of invoking Tcl/Tk from inside of ANSYS.

5.1.1 Tcl Shell

The Tcl shell can be invoked by using ~tcl. This is the Tcl shell with the addition of custom ANSYS commands for calling the ANSYS API from the shell.

5.1.2 Tcl/Tk Shell (wish)

The Tcl/Tk Shell can be invoked by using ~tk. This is the Tcl/Tk shell (wish) with the addition of custom ANSYS commands for calling the ANSYS API from the shell.

5.1.3 Enhanced UIDL

Enhanced UIDL can be invoked by using ~eui. This creates an evironment that is used by the Enhanced UIDL code of ANSYS. It pulls in the [incr Tcl]/[incr Tk] packages as well.

5.2 Calling the ANSYS API

The following gives an overview of functionality see 5.4 Command Descriptions for the specific use of a command.

5.2.1 Core ANSYS

The API to the core ANSYS functionality is provided in the following commands: ans_sendcommand, ans_getvalue, and ans_getvector. This provides a large percentage of the functionality of ANSYS.

5.2.2 ANSYS Graphics

The API to the ANSYS Graphics is provided in the following command: ans_sendcommand. This allows manipulation of the graphics via the ANSYS graphic related commands.

5.2.3 ANSYS GUI

The API to the ANSYS Graphics is provided in the following command: ans_sendcommand. Many of the custom widgets can be invoked by using the /UI command. UIDL widgets can be invoked by sending its Fnc_Name. To make a UIDL widget modal to Tcl/Tk you should use uidl::callFnc or code similar to it.

5.2.4 ANSYS Graphical Picking

The API to the ANSYS Graphics is provided in the following commands:

ans_pick_entity, ans_pick_entitydone, ans_pick_entitydump, ans_pick_entitydumpdup, ans_pick_entityinfo, ans_pick_entityinqr, ans_pick_entitypickall, ans_pick_entityrange, ans_pick_entityreset, ans_pick_entityrestart, ans_pick_xyz, ans_pick_xyzadd, ans_pick_xyzdone, ans_pick_xyzdump, ans_pick_xyzinfo, ans_pick_xyzinqr, ans_pick_xyzreset. There is no error trapping at this time for these commands so they must be used with caution. For example if you start picking and don't stop it and try to use one of the UIDL picking menus it will not work.

5.3 Data storage

The maintenance of data created in Tcl/Tk may need to be kept in the ANSYS database. The best way to do this is to create a relationship of Tcl/Tk variables with ANSYS array positions. A working model of this can be found in the file $ANSYS5x_DIR/lib/Euidl1.0/euidl.tcl. ANSYS development personnel must use the mechanisms provided for storing data from Tcl/Tk to the ANSYS database. Application developers should use it as a skeleton for setting up there own area in the database.

5.4 Command Descriptions

NAME

ans_cleardbcb

SYNOPSIS

ans_cleardbcb *tclProc*

DESCRIPTION

This command will register *tclProc* such that when the /CLEAR command is issued in ANSYS it will evaluate *tclProc* before after clearing the database in ANSYS. The ans_cleardbcb command should be issued only one time.

NAME

ans_flushevents

SYNOPSIS

ans_flushevents *boolean*

DESCRIPTION

This command will cause the ANSYS event loop to be flushed from Tcl/Tk when Tcl/Tk is blocking. This allows UIDL or custom widgets to be invoked from a Tcl/Tk modal dialog and have them function. The value of *boolean* must be set to 1 before blocking and set to 0 after blocking is finished.

NAME

ans_evalexpr

SYNOPSIS

ans_evalexpr *APDL expression*

DESCRIPTION

This command takes an *APDL expression* for evaluation. It returns the evaluated value of the *APDL expression* to the interpreter.

EXAMPLE

set nd_${i}_x [ans_evalexpr nx($I)]

if the the value of A and B are known in APDL the sum can be gotten:

set sumAB [ans_evalexpr A+B]

NAME

ans_getvalue

SYNOPSIS

ans_getvalue *ansys*GetConstruct*

DESCRIPTION

This command takes an *ansys*GetConstruct* which is fields 3 through field 8 of the ANSYS *GET command. It returns the value of the *ansys*GetConstruct* to the interpreter.

EXAMPLE

```
set ansRev [ans_getvalue active,,rev]

set postUSUM(1) [ans_getvalue node,1,u,sum]
```

NAME

ans_getvector

SYNOPSIS

ans_getvector *ansysAPDLArray*

DESCRIPTION

This command will return a list of lists that contain the data from *ansysAPDLArray*. The last list contains the number of rows columns and planes in the *ansysAPDLArray*. There will be planes + 1 lists returned. These lists from 0 to end –1 will then contain column lists that contain the column data and will have row elements.

EXAMPLE

Input

```
x=2
y=2
z=2
*dim,dp_arry,,x,y,z
*do,i,1,x,1 !{
*do,j,1,y,1 !{
*do,k,1,z,1 !{
dp_arry(i,j,k)=i*100+j*10+k
*enddo !}
*enddo !}
*enddo !}
*stat,dp_arry
~tcl,'puts "tcl status for dp_arry:\n[ans_getvector dp_arry]"'
```

Output

```
PARAMETER STATUS- DP_ARRY   (  12 PARAMETERS DEFINED)
                (INCLUDING     3 INTERNAL PARAMETERS)

    LOCATION              VALUE
      1      1      1    111.000000
      2      1      1    211.000000
```

```
1     2     1   121.000000
2     2     1   221.000000
1     1     2   112.000000
2     1     2   212.000000
1     2     2   122.000000
2     2     2   222.000000
tcl status for dp_arry:
{{111.0 211.0} {121.0 221.0}} {{112.0 212.0} {122.0 222.0}} {2 2 2}
```

NAME

ans_loadhelp

SYNOPSIS

ans_loadhelp *ansysHelpString*

DESCRIPTION

This command will pass *ansysHelpString* to the ANSYS help interpreter to display the requested help.

NAME

ans_pick_entity – Begin selection of entities in the ANSYS graphic window.

SYNOPSIS

ans_pick_entity *entityType entitySelection rubberbandMode duplicates order min max var*

DESCRIPTION

This command will begin selection of the entities specified by *entityType* using the additional arguments to effect the selection. Please see the ans_pick_enitityinqr command for the valid options for *entityType*, *enititySelection*, *rubberbandMode*, *duplicates*, *order*, *min*, and *max*.

var - *th*e variable *var* is linked as an array so that each time an entity is chosen it will be updated. The *var* array is also used for other commands for return values. The *var* array will look as follows with possible values:

var(EventType)	-	0	- motion
		1	- button press
		2	- button release
		3	- key press
		4	- key release
		99	- cancel
var(MouseButton)	-	0	- none
		1	- left mouse button
		2	- middle mouse button
		3	- right mouse button
		n	- ascii key for key event
var(ButtonMask)	-	n	- integer representing the button mask for the event
var(ScreenX)	-	n	- the graphic window screen x coordinate
var(ScreenY)	-	n	- the graphic window screen y coordinate
var(EntityNumber)	-	n	- the entity number of the currently selected entity
var(EntityType)	-	1	- node
		2	- element

3 - keypoint

4 - line

5 - area

6 - volume

7 - trace point

8 - component

9 - graph point

var(GlobalX) - d - double value for the global Cartesian X value of the entity hotspot

var(GlobalY) - d - double value for the global Cartesian Y value of the entity hotspot

var(GlobalZ) - d - double value for the global Cartesian Z value of the entity hotspot

var(WpX) - d - has no meaning in this context

var(WpY) - d - has no meaning in this context

NAME

ans_pick_entitydone – Finish selection of entities in the ANSYS graphic window.

SYNOPSIS

ans_pick_entitydone

DESCRIPTION

This command will finish selection of entities.

NAME

ans_pick_entitydump – Sets the *var*(entityList) where *var* is from ans_pickentity.

SYNOPSIS

ans_pick_entitydump

DESCRIPTION

This command will set *var*(entityList) to a list for each entity selected where the list has as the first element the entity id and as the second element the type of entity.

NAME

ans_pick_entitydumpdup – Sets the *var*(duplicateList) where *var* is from ans_pickentity.

SYNOPSIS

ans_pick_entitydumpdup

DESCRIPTION

This command will set *var*(duplicateList) to a list for each duplicate entity selected where the list has as the first element the entity id and as the second element the type of entity. This is used for a duplicate list of entities for the current pick, i.e. picking keypoints and several are at the same screen position.

NAME

ans_pick_entityinfo – Provides information to the current entity picking operation.

SYNOPSIS

ans_pick_entityinfo *optionKey optionValue*

DESCRIPTION

This command is used to change the behavior of the entity picking operation.

OptionKey - 0 - Selection mode

optionValue - 0 - unselect mode

1 - select mode

1 - Selection style

optionValue - 0 - single selection

1 - box selection

2 - polygon selection

3 - circle selection

4 - loop selection

NAME

ans_pick_entityinqr – Returns information from the current entity picking operation.

SYNOPSIS

ans_pick_entityinqr *optionKey*

DESCRIPTION

This command is used to return information about the entity picking operation.

OptionKey - 0 - Selection mode

return value - 0 - unselect mode

1 - select mode

1 - Selection style

return value - 0 - single selection

1 - box selection

2 - polygon selection

3 - circle selection

4 - loop selection

3 - Entity types for selection, where a combination is multiple entities

return value - 1 - nodes

2 - elements

4 - keypoints

8 - lines

16 - areas

32 - volumes

64 - trace points

128 - components

256 - graph points

4 - Selected state of the entities being selected

return value - 1 - only selected entities can be picked.

-1 - only unselected entities can be picked

0 - both selected and unselected entities can be picked.

5 - Rubberband Style

return value - 1 - lines
2 - rectangle (corner - opp/corner)
3 - circle (center - radius)
4 - annulus (center - r1 - r2)
5 - partial annulus
6 - rectangle (center - corner)
7 - circle (on circ - on circ)
8 - square (center - on edge)
13 - 3 equal-sided polygon
14 - 4 equal-sided polygon
15 - 5 equal-sided polygon
16 - 6 equal-sided polygon
17 - 7 equal-sided polygon
18 - 8 equal-sided polygon
101 - polyhedren
102 - block
103 - cylinder-solid
104 - cylinder-hollow
105 - partial cylinder
106 - block (center - corner)
107 - cylinder-solid (on circ - on circ)
108 - cone
113 - 3 equal-sided polyhedren
114 - 4 equal-sided polyhedren
115 - 5 equal-sided polyhedren
116 - 6 equal-sided polyhedren
117 - 7 equal-sided polyhedren
118 - 8 equal-sided polyhedren
201 - electric inductor
202 - electric capacitor

203 - electric resistor
204 - electric circuit cr1
205 - electric circuit cr2
206 - electric circuit cr3
207 - electric circuit cr4
208 - electric mutual inductor
209 - torsional spring
210 - simple diode
211 - zener diode
212 - linear spring
213 - linear damper
214 - torsional damper
215 - EMT transducer
216 - piezoelectric crystal
217 - lumped mass

6 - Allow duplicate picks
return value 0 - Not Allowed
1 - Allowed
7 - Picked entity ordering
return value - 0 - Unordered
1 - Ordered
8 - Minimum entities required
return value - n - integer of minimum required
9 - Maximum entities required
return value - n - integer of maximum required
10 - Number of entities currently selected
return value - n - integer of selected entities
11 - Number of duplicate entities for current selection
return value - n - integer of selected duplicate entities
12 - Entity ID of currently selected entity
return value - n - id of currently selected entity
13 - Entity type for the currently selected entity
return value - n - entity type of currently selected entity

14 - Number of entity types picked

return value - n - integer of number of entity types picked.

15 - Bitmask of entity types picked

return value - n - sum of entity types picked see *optionKey* 3 for the bits

NAME

ans_pick_entitypickall – Selects all of the entities for the current entity type selection.

SYNOPSIS

ans_pick_entitypickall

DESCRIPTION

This command is used to select all entities of the entity type specified on the ans_pick_entity command.

NAME

ans_pick_entityrange – Selects the entities for a given range.

SYNOPSIS

ans_pick_entitypickall *entityType rangeType rangeStart* ?args …?

DESCRIPTION

This command is used to select a range of entities of type *entityType*.

entityType - See the ans_pick_entityinqr *optionKey* 3 for bitmask values.

rangeType - This can be 0 for a list of entities or 1 for a range of entities.

rangeStart - The beginning of the range of data.

EXAMPLE

ans_pick_entityrange 8 0 1 3 4 5 20 30 40 50

ans_pick_entityrange 8 1 1 50 3

NAME

ans_pick_entityreset – Reset the picking operation.

SYNOPSIS

ans_pick_entityreset

DESCRIPTION

This command is used to reset the pick list to empty.

NAME

ans_pick_entityrestart – Restart the picking operation.

SYNOPSIS

ans_pick_entityrestart

DESCRIPTION

This command is used to reset the pick list to empty and checks for available enities as specified on the ans_pick_entity command.

NAME

ans_pick_xyz – Begin locational picking in the ANSYS graphic window.

SYNOPSIS

ans_pick_xyz *pickType rubberbandMode min max var*

DESCRIPTION

This command will begin picking points on the working plane or screen as specified by *pickType*. using the additional arguments to effect the selection. Please see the ans_pick_xyzinqr command for the valid options for *rubberbandMode*, *min*, and *max*.

pickType - 0 - Working plane location picking

1 - Screen location picking

var - *th*e variable *var* is linked as an array so that each time an entity is chosen it will be updated. The *var* array is also used for other commands for return values. The *var* array will look as follows with possible values:

var(EventType) - 0 - motion

1 - button press

2 - button release

3 - key press

4 - key release

99 - cancel

var(MouseButton) - 0 - none

1 - left mouse button

2 - middle mouse button

3 - right mouse button

n - ascii key for key event

var(ButtonMask) - n - integer representing the button mask for the event

var(ScreenX) - n - the graphic window screen x coordinate

var(ScreenY) - n - the graphic window screen y coordinate

var(EntityNumber) - no meaning in this context

var(EntityType) - no meaning in this context

var(GlobalX) - d - double value for the global Cartesian X
var(GlobalY) - d - double value for the global Cartesian Y
var(GlobalZ) - d - double value for the global Cartesian Z
var(WpX) - d - double value for the working plane X value
var(WpY) - d - double value for the working plane Y value

NAME

ans_pick_xyzadd – Add a given XYZ location to the picked set.

SYNOPSIS

ans_pick_xyzadd *x y z*

DESCRIPTION

This command will add the location (x,y,z) to the picked set and also highlight the location on the screen.

NAME

ans_pick_xyzdone – Finish location picking.

SYNOPSIS

ans_pick_xyzdone

DESCRIPTION

This command will finish the location picking operation.

NAME

ans_pick_xyzdump – Sets the *var*(xyzList) where *var* is from ans_pickxyz.

SYNOPSIS

ans_pick_xyzdump

DESCRIPTION

This command will set *var*(xyzList) to list for each location picked where the list has as the first element the x-location, the second element the y-location, the third element the z-location.

NAME

ans_pick_xyzinfo – Provides information to the current location picking .

SYNOPSIS

ans_pick_xyzinfo *optionKey optionValue*

DESCRIPTION

This command is used to change the behavior of the location picking operation.

OptionKey - 0 - Selection mode

optionValue - 0 - unselect mode

1 - select mode

NAME

ans_pick_xyzinqr – Returns information from the current entity picking operation.

SYNOPSIS

ans_pick_xyzinqr *optionKey*

DESCRIPTION

This command is used to return information about the entity picking operation.

OptionKey - 0 - Selection mode

return value - 0 - unselect mode

1 - select mode

5 - Rubberband Style

return value - 1 - lines

2 - rectangle (corner - opp/corner)

3 - circle (center - radius)

4 - annulus (center - r1 - r2)

5 - partial annulus

6 - rectangle (center - corner)

7 - circle (on circ - on circ)

8 - square (center - on edge)

13 - 3 equal-sided polygon

14 - 4 equal-sided polygon

15 - 5 equal-sided polygon

16 - 6 equal-sided polygon

17 - 7 equal-sided polygon

18 - 8 equal-sided polygon

101 - polyhedren

102 - block

103 - cylinder-solid

104 - cylinder-hollow

105 - partial cylinder

106 - block (center - corner)

107 - cylinder-solid (on circ - on circ)

108 - cone
113 - 3 equal-sided polyhedren
114 - 4 equal-sided polyhedren
115 - 5 equal-sided polyhedren
116 - 6 equal-sided polyhedren
117 - 7 equal-sided polyhedren
118 - 8 equal-sided polyhedren
201 - electric inductor
202 - electric capacitor
203 - electric resistor
204 - electric circuit cr1
205 - electric circuit cr2
206 - electric circuit cr3
207 - electric circuit cr4
208 - electric mutual inductor
209 - torsional spring
210 - simple diode
211 - zener diode
212 - linear spring
213 - linear damper
214 - torsional damper
215 - EMT transducer
216 - piezoelectric crystal
217 - lumped mass

8 - Minimum locations required

return value - n - integer of minimum required

9 - Maximum locations required

return value - n - integer of maximum required

10 - Number of locations currently picked

return value - n - integer of picked locations

NAME

ans_pick_xyzreset – Reset the picking operation.

SYNOPSIS

ans_pick_xyzreset

DESCRIPTION

This command is used to reset the pick list to empty.

NAME

ans_sendcommand – Send an ANSYS command to the ANSYS program for evaluation.

SYNOPSIS

ans_sendcommand *ansysCommand* ?arg …?

DESCRIPTION

This command will pass *ansysCommand* to the ANSYS command interpreter to be evaluated. The return value will be 1 (evaluation caused a note), 2 (evaluation caused a warning), 3 (evaluation caused an error).

NAME

ans_senderror

SYNOPSIS

ans_senderror *errorLevel string*

DESCRIPTION

This command will cause ANSYS to display *string* at the specified *errorLevel*.

The *errorLevel* may have a value of 1 (note), 2 (warning), 3 (error), or 4 (fatal).

NAME

ans_writeout

SYNOPSIS

ans_writeout *string*

DESCRIPTION

This command will write *string* to the ANSYS output. If /NOPR is turned on in ANSYS then nothing is written to the output. The string is buffered until a newline “\n” is found.