ANSYS Graphical User Interface Programming with Tcl/Tk

Presented By

CAE Associates, Inc.

© 2013 CAE Associates
Presentation Topics

- Introduction
  - What is Tcl/Tk?
  - Advantages and Disadvantages of Tcl/Tk

- Tcl/Tk in ANSYS
  - Basic command structure
  - Calling user Tcl/Tk code from ANSYS
  - Access to the ANSYS API
  - GUI component creation

- Example Application
  - Parametric Design Entry Dialog Box

- Tcl/Tk Tools and Information
  - Visual Development Tools
  - Sources of Information
Introduction What is Tcl/Tk?

- Tcl/Tk is pronounced “tickle-tee-kay”.
  - Tcl is a universal scripting language.
  - Tk is a Graphical User Interface (GUI) toolkit and widgets based in Tcl.
- Developed by John Ousterhout at the University of California, Berkeley in 1989.
- Custom Tcl/Tk interpreters have been compiled into the ANSYS analysis environment since Revision 5.5.
- Some existing ANSYS Tcl/Tk components:
  - Material Modeler
  - Contact Wizard
  - Solution Controls
  - Time History Variable Viewer
  - Entire GUI (Revision 6.1)
Introduction Advantages of Tcl/Tk

- **Price:** It’s Free !!!
- **Learning Curve:**
  - Easy learning curve compared with traditional software languages such as C++ and Fortran.
  - Interpreted (not compiled) so effects of modifications are immediately realizable.
- **Flexibility:**
  - Wide variety of tools to create entry boxes, canvases, scrollable forms, labels, etc.
  - Create own tools and procedures.
- **Compatibility:**
  - Cross-platform compatible.
Introduction Disadvantages of Tcl/Tk

- **Sparse ANSYS Documentation:**
  - ANSYS documentation/training is sparse compared to UIDL and APDL.
  - Limited technical support since GUI programming is considered a *non-standard* use of the software.

- **Small ANSYS User Base:**
  - ANSYS implementation is recent so user base is limited.
  - Very few user defined scripts and libraries exist in the public domain.

- **Performance:**
  - To maintain platform independence, Tcl/Tk is not compiled, rather it is an *interpreted* language and implementation is *on the fly*.
  - This tends to cause some degradation of speed particularly when trying to access ANSYS over a network connection.
Tcl/Tk in ANSYS Command Structure

- Tcl/Tk syntax is simple — similar to sh, C, and Lisp.
- A command is formed by words separated by white space.
- Dollar sign ($) substitutes the value of a variable.
- Square brackets executes a nested command. Result from cmd2 passed as argument to cmd1.
- Double quotes group words into a single argument.
- Curly braces group words into a single argument, however, elements within the braces are not interpreted.

<table>
<thead>
<tr>
<th>Tcl Expression</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>set x 2</td>
<td>x=2</td>
</tr>
<tr>
<td>set y $x</td>
<td>y=2</td>
</tr>
<tr>
<td>set z [expr $x+$y]</td>
<td>z=4</td>
</tr>
<tr>
<td>set t “z is $z”</td>
<td>t=“z is 4”</td>
</tr>
<tr>
<td>set t {z is $z}</td>
<td>t=“z is $z”</td>
</tr>
</tbody>
</table>
There are various methods for calling your Tcl/Tk code from within ANSYS.

- Tcl shell used for running scripts without GUI features:
  \[ \sim\text{tcl},'\text{source filename}' \]

- Tcl/Tk shell for using GUI features:
  \[ \sim\text{tk},'\text{source filename}' \]

- Enhanced UIDL for including object oriented and some ANSYS objects:
  \[ \sim\text{eui},'\text{source filename}' \]

- From UIDL menus

```
:N Fnc_MyTclFunc
:S 0, 0, 0
:T Command
:A My Tcl Function
:Inp_P
:Cmd_\) \sim\text{eui},'\text{tcl/tk command}'
:E END
```
### ANSYS API Core Functionality

- Access to the core ANSYS functionality is provided via `ans_*` series of commands.
- Some frequently used commands:

<table>
<thead>
<tr>
<th>Functionality</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ans_sendcommand</strong></td>
<td><code>ansysCommand</code></td>
</tr>
<tr>
<td><code>set n1 1</code></td>
<td>Passes a command to ANSYS for processing</td>
</tr>
<tr>
<td><code>set n2 2</code></td>
<td></td>
</tr>
<tr>
<td><code>ans_sendcommand n,,$n1,$n2</code></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Functionality</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ans_getvalue</strong></td>
<td><code>ansys*GetConstruct</code></td>
</tr>
<tr>
<td><code>set ansRev [ans_getvalue ACTIVE,,REV]</code></td>
<td>*GET an ANSYS value</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Functionality</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ans_getvector</strong></td>
<td><code>ansysAPDLArray</code></td>
</tr>
<tr>
<td>! APDL Command</td>
<td>Returns an ANSYS array</td>
</tr>
<tr>
<td>! Tcl Commands</td>
<td></td>
</tr>
<tr>
<td>*dim,apdlArr,,2</td>
<td></td>
</tr>
<tr>
<td>apdlArr(1) = 1,2</td>
<td></td>
</tr>
</tbody>
</table>
Tk provides numerous widgets for GUI creation.

Example widgets:

- **toplevel** `.dlgMyDialog`
- **combobox** `.cmbFont`
- **button** `.btnOK`
- **checkbox** `.chbStyles`
- **labeledframe** `.frmPrintRange`
Some of the most common widgets used are the:

**Label:**

```python
set myGUI(title) [label $myGUI(frame).lblTitle \ -text "Create Node at Specified X and Y value"]
set myGUI(xLabel) [label $myGUI(frame).lblX \ -text "X: "]
set myGUI(yLabel) [label $myGUI(frame).lblY \ -text "Y: "]
```

**Entry Box:**

```python
set myGUI(entry.x) [entry $myGUI(frame).entXcoord \ -textvariable xcoord -bg white]
set myGUI(entry.y) [entry $myGUI(frame).entYcoord \ -textvariable ycoord -bg white]
```

**Button:**

```python
set myGUI(buttonOK) [button $myGUI(frame).btnOK \ -text OK -bg grey -command {
  ans_sendcommand "*set, _nx, $xcoord"
  ans_sendcommand "*set, _ny, $ycoord"
  ans_sendcommand "*set, BUTTON, 2"
  ans_sendcommand "n, _nx, _ny"
  destroy $myGUI(parent)
}]
```

- **Use the** `-text` **option**
- **Use the** `-textvariable` **option to set the variable name associated with the entry**
- **The** `-text` **option specifies a label for the button**
- **The** `-command` **option specifies a series of commands to perform once the button is pressed.**
Example Applications Parametric Design

- Create customized GUI for parametric analysis.
- Users execute an ANSYS macro which has calls to Tcl/Tk code.

  ```
  ~/eui,'source filename'
  ```

- Users provide input for creating model and running the solution.
- All necessary analysis steps are predefined and transparent to the user.
- Freely-available, high-quality application development environment.
- Written entirely in Tcl/Tk and generate pure Tcl/Tk code.

Visual Gypsy
http://www.prs.de/int/index.html

Visual Tcl
http://vtcl.sourceforge.net/
**Other Tcl/Tk Tools** Sources of Information

**ANSYS Information:**
- Chapters 5-7 of the *ANSYS GUI Style Guide*.
- ANSYS 2002 Users’ Conference

**General Tcl/Tk Information:**
- Tcl Developers Exchange at
  
  **www.scriphtics.com**

- ActiveState – developers of Tcl/Tk tools and allied with ANSYS
  
  **www.activestate.com**