Astronomical Data Analysis Software and Systems VI
ASP Conference Series, Vol. 125, 1997
Gareth Hunt and H. E. Payne, eds.

Tcl- and [Incr tcl]- Based Applications for Astronomy and the Sciences

Nicholas M. Elias II

United States Naval Observatory, Navy Prototype Optical Interferometer, 3450 Massachusetts Avenue NW, Washington, DC 20392-5420, e-mail: nme@fornax.usno.navy.mil

Abstract. Tcl is a shell/script language used to create both instrumental control systems and interactive data reduction programs. It has extensions for creating GUI (Tk) and object-oriented ([incr tcl]) applications. Several Tcl-based tools that may be used for astronomical and other scientific applications have been created and are discussed. One example is *ptcl*, which registers PGPLOT functions as Tcl commands, creating a powerful interactive plotting package. The Tcl-astronomy WWW home-page and majordomo mailing list server are also described.

1. Introduction

Tcl is a general-purpose shell and script language. It is maintained by J. K. Ousterhout,[1] and is available free of charge over the World Wide Web.[2]

Tcl consists of a powerful set of "core" commands that look like a cross between C functions and (t)csh commands. This core includes math functions, variables, associative arrays, string and list commands, flexible parsing/substitution, program control (for, while, if, etc.), regular expressions, I/O, and error handling. New Tcl commands can be added either by "registering" C/FORTRAN functions or using the "proc" Tcl core command. Tcl can also access UNIX shell commands without awkward escape sequences. Therefore, Tcl may be used as "glue" to create exciting new software packages.

Tcl has some advantages over compiled languages such as FORTRAN, C, or C++. For example, programming ideas can be tested "on the fly" (while the application is running), decreasing development time. Also, Tcl installs on any UNIX workstation that has an ANSI-C compiler (versions for MS-Windows and Macintosh exist as well), making Tcl-based applications very portable. In addition, Tcl can be bundled with other software with no restrictions or charges.

2. The Most Common Tcl Extensions, Tk and [Incr tcl]

Most scientists use X-Windows based window managers to interact with the UNIX operating systems on their workstations. Creating graphical user inter-

[1] http://www.sunlabs.com:80/people/john.ousterhout

[2] http://www.sunlabs.com/research/tcl

faces (GUIs) in C is straightforward but very time consuming, especially in the debugging stage. J. K. Ousterhout's answer to this problem is the Tcl extension called Tk,[3] an X-Windows toolkit. This toolkit consists mainly of window widgets registered as Tcl commands. With Tk, all the annoying details of X-Windows are hidden from the programmer, allowing him/her to create GUIs in minutes to hours instead of hours to days. More complicated "mega-widgets" (typically combinations of the standard widgets) can be created by writing C code, if desired. Like Tcl, Tk installs on any UNIX workstation (it must have X-Windows, of course) that has an ANSI-C compiler, and Tcl/Tk scripts are totally portable (no machine dependent C code is necessary).

Object-oriented programming systems (OOPS) are becoming more and more prevalent because of their many advantages, such as configurable class templates, inheritance, and "members-only" manipulation of data (encapsulation). [Incr tcl][4] is an object-oriented extension of Tcl, created by M. J. McLennan.[5] In addition to the above advantages, [incr tcl] is useful for grouping a large number of Tcl commands into classes, making a command-line application much more manageable.

3. Tcl-Based Tools and Software

Many software libraries have been created over the years by scientists for scientists, and have become standards in the community. These packages can be made interactive using Tcl, thus extending their usefulness. For example, PGPLOT[6] (the Caltech plotting package written by T. J. Pearson) library functions have been registered as Tcl commands in the *ptcl*[7] package. Also, HDS[8] (the Hierarchical Database System of the Starlink Project) functions have been registered as Tcl commands in the *htcl*[9] package.

A sample *ptcl* script and its corresponding PGPLOT FORTRAN subroutine are shown in Figure 1. The Tcl script and FORTRAN code are very similar, which means that someone who is already familiar with PGPLOT can learn *ptcl* quickly. Also, recall that Tcl scripts can be modified and tested while an application is still running, which means that the Tcl procedure quad_plot can be modified more quickly than the FORTRAN subroutine QUAD_PLOT.

Software for the Navy Prototype Optical Interferometer (NPOI) laser metrology system[10] has been created using [incr tcl], *ptcl* and *htcl*. The general structure of these programs (bottom to top) is: low-level C code, mid-level C code, C code that registers mid-level functions as Tcl commands, and [incr

---

[3] http://www.sunlabs.com/research/tcl

[4] http://www.tcltk.com/itcl/index.html#moreInfo

[5] http://www.tcltk.com/itcl/mmc.html

[6] http://astro.caltech.edu/~tjp/pgplot

[7] ftp://fornax.usno.navy.mil/dist/ptcl/ptcl.html

[8] http://star-www.rl.ac.uk

[9] ftp://fornax.usno.navy.mil/dist/htcl/htcl.html

[10] http://aries.usno.navy.mil/ad/npoi/npoi.html

```
proc quad_plot {x} {

  set n [llength $x]

  set y ""
  foreach x2 $x {
    lappend y [expr $x2*$x2]
  }

  pgbeg 0 /xs 1 1

  pgsci 1
  pgsch 1.3

  set xmin [lindex $x 0]
  set xmax [lindex $x [expr $n-1]]
  set ymin [lindex $y 0]
  set ymax [lindex $y [expr $n-1]]
  pgenv $xmin $xmax $ymin $ymax 0 0

  pglab x y "Quad Plot"

  pgsci 2
  pgpt $n $x $y 17

  pgsci 4
  pgline $n $x $y

  pgend

}
```

```
      SUBROUTINE QUAD_PLOT( N, X )

      INTEGER*4 N
      REAL*4 X(N),Y(N)

      DO 10 I=1,N
        Y(I)=X(I)*X(I)
   10 CONTINUE

      CALL PGBEG( 0, "/xs", 1, 1 )

      CALL PGSCI( 1 )
      CALL PGSCH( 1.3 )

      CALL PGENV( X(1), X(N), Y(1),
     :     Y(N), 0, 0 )

      CALL PGLAB( "x", "y",
     :     "Quad Plot" )

      CALL PGSCI( 2 )
      CALL PGPT( N, X, Y, 17 )

      CALL PGSCI( 4 )
      CALL PGLINE( N, X, Y )

      CALL PGEND

      RETURN

      END
```

Figure 1. A simple *ptcl* procedure to plot the square of an array. For the sake of comparison, a FORTRAN subroutine using PGPLOT subroutines is also shown.

tcl] classes corresponding to each type of data (logs, raw data, averaged data, configuration information, etc.). The [incr tcl] classes can be accessed directly from the Tcl command line or from GUIs. There are two main programs in this package, FAKE (creates simulated data) and INCHWORM (laser interferometer and environmental sensor analysis). Both programs use the same C code, Tcl commands, and [incr tcl] classes; only the GUIs are different. A library of Tcl scripts are also included.

4. Tcl and the World Wide Web

New "bytecode" languages, such as Java, are now being used to create exciting Web-based applets. Tcl can now be considered a "Web language", since Tcl/Tk bytecode "plug-ins"[11] have been created for Netscape, i.e., Netscape can be used to run Tcl/Tk applet scripts.

A Tcl-astronomy homepage on the World Wide Web is now being maintained by the author. *ptcl* and *htcl* can be obtained here, and links to other Tcl pages are provided as well. In addition, De Clark (de@ucolick.org) maintains a majordomo mailing list manager called "tclastro"; check the Tcl-astronomy home page for directions on how to subscribe.

---

[11] http://www.sunlabs.com/research/tcl/plugin