New perltk GUI tools for *Antelope*

June, 2008

Antelope User Group Meeting

Skamania Lodge, WA

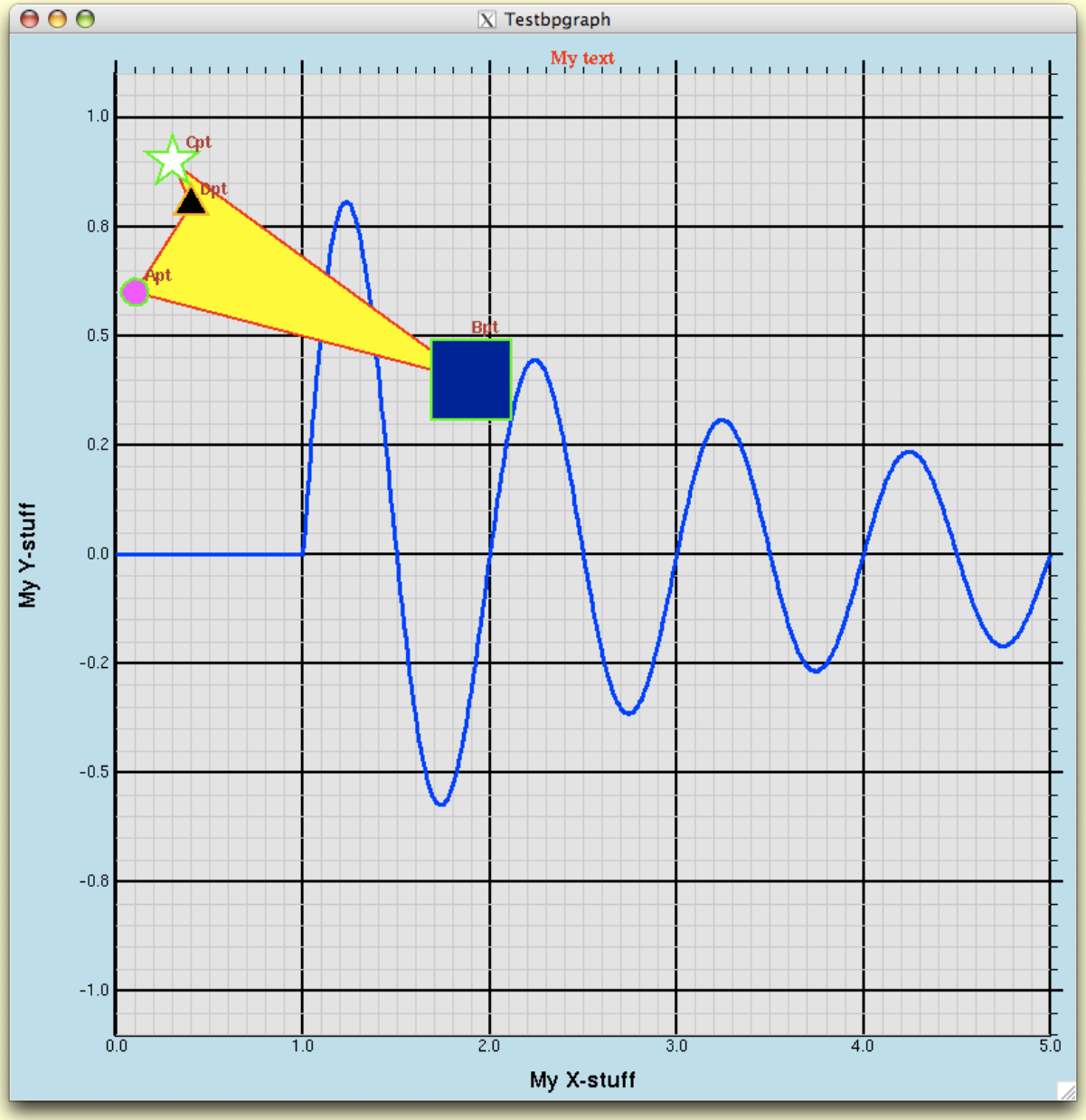
BRTT

BOULDER REAL TIME TECHNOLOGIES, INC.

In 4.10 all of the tcl/tk brttplot graphics widgets have been converted to perltk and added as a dynamic perl package

```
BPLOT(3P)                          Perl Extensions Commands                          BPLOT(3P)

NAME
       bpviewport,  bpaxes,  bpgrid, bptext, bppolyline, bppolypoint, bpmap, bptrace -
       BRTT perltk canvas item extensions

SYNOPSIS
       use TK::Canvas
       use Tk::Bplot

       $canvas->create('bpviewport', viewportname, x, y, ?option, value, ...?)

       $canvas->create('bpaxes', viewportname, ?option, value, ...?)

       $canvas->create('bpgrid', viewportname, ?option, value, ...?)

       $canvas->create('bptext', viewportname, textstring, x, y, ?option, value, ...?)

       $canvas->create('bppolyline', viewportname, ?option, value, ...?)

       $canvas->create('bppolypoint', viewportname, ?option, value, ...?)

       $canvas->create('bpmap', viewportname, ?option, value, ...?)

       $canvas->create('ç', viewportname, ?option, value, ...?)
```

BRTT



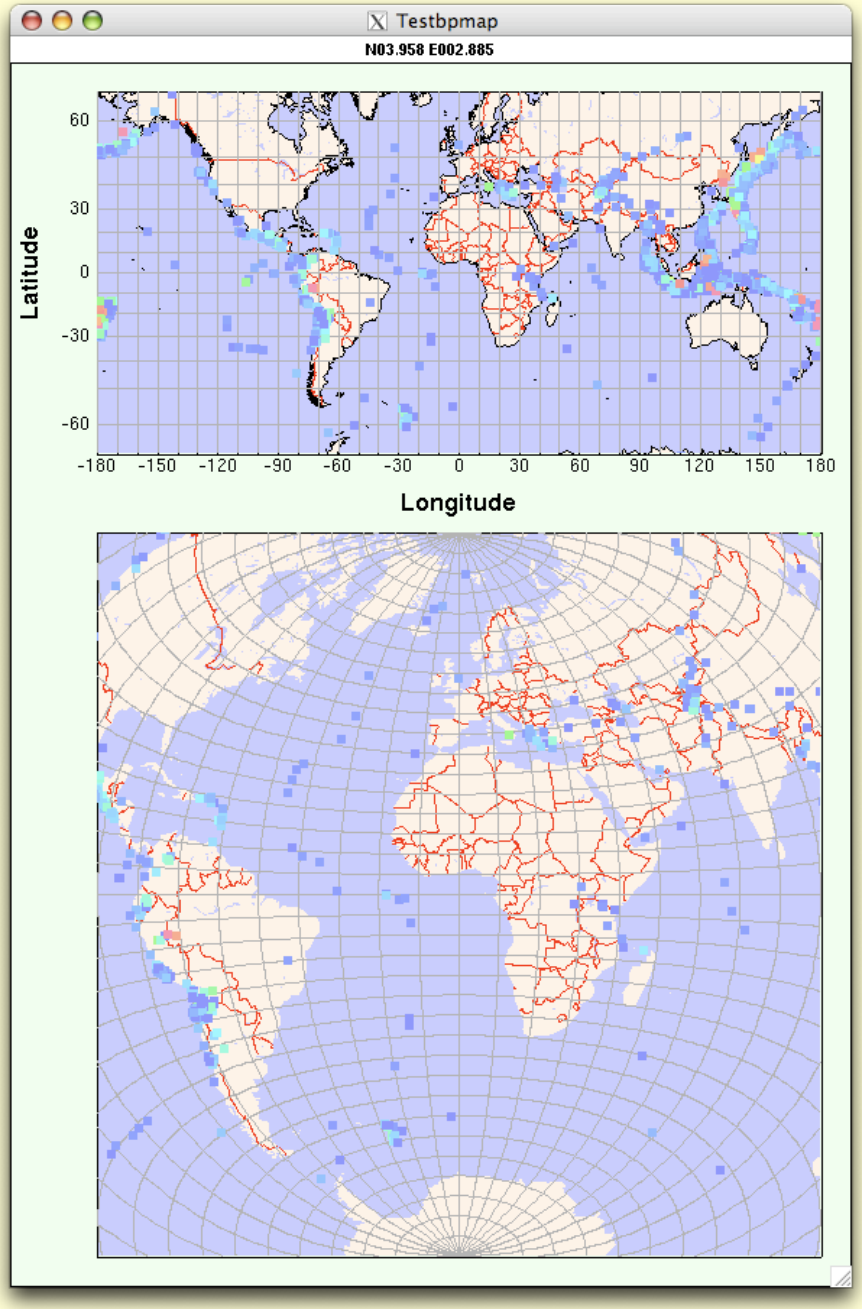
BRTT

June 2008

- Like with tcl/tk version, designed as new canves items
- New canvas items include **`bpviewport`**, **`bpaxes`**, **`bpgrid`**, **`bptext`**, **`bppolyline`**, **`bppolypoint`**, **`bpmap`** and **`bptrace`**
- New perl extension **`Vector`** used to feed large vectors into polyline and polypoint items (see **`vector(3p)`**)
- High resolution PostScript output
- Support for canvas bindings

BRTT

```
#    now we are going to create a vector, open
#    a database with some origins, read the origin
#    lats, lons and depths, and fill in the vector

our $events = vector_create ;

my @db = dbopen 'gsn', 'r' ;
my @dbe = dblookup @db, 0, 'event', 0, 0 ;
my @dbo = dblookup @db, 0, 'origin', 0, 0 ;

our @db = dbjoin @dbe, @dbo, 'prefor#orid' ;

my $nrecs = dbquery @db, 'dbRECORD_COUNT' ;

for ($db[3] = 0; $db[3] < $nrecs; $db[3]++) {
     my ($lat, $lon, $depth) = dbgetv @db, 'lat', 'lon', 'depth' ;
     my $color = setcolor $depth ;
     vector_append ( $events, -1, $lon, $lat, sprintf ( '{f=%s}', $color ) ) ;
}

#    now we create polypoint items to show the origins
#    in each map

$canvas->create ( 'bppolypoint', 'vpm',
          -vector => $events,
          -symbol => 'square',
          -fill => 'blue',
          -outline => '',
          -size => 3,
     ) ;
```



BRTT

June 2008

Caveats and plans

- Stick to canvas bindings - canvas item bindings do not work
- Working on **`dbpick`**-style graphics and bindings for waveform/arrival display and manipulation
- Plans for new canvas item for animated display of real-time ORB waveform packets
- Plans to rewrite **`dbevents`**, **`dbpick`** and **`orbmonrtd`** using these and new similar widgets