

# Antelope Development

---

a whirlwind tour

Daniel Quinlan  
BRTI, Inc.

# Antelope provides

- \* database operations
- \* time calculations
- \* parameter file interface
- \* waveform i/o
- \* response calculations
- \* travel time calculation
- \* orb interface
- \* packet stuff/unstuff
- \* miscellaneous utilities

- \* Basic functions are coded in c
- \* Relatively complete documentation in man pages
- \* interfaces to these functions are also provided in other languages.

# Antelope languages

\* c

\* perl

BRTT

\* tcl/tk

\* shell

---

\* matlab

Kent

\* pfp

---

\* python (limited distribution)

Passcal

# Too many choices?

- \* Best all around choice: perl
  - \* most general
  - \* best string manipulation
  - \* powerful
  - \* scripting language => easy
  - \* widely used

# Can't use perl? Don't like perl?

- \* heavy duty computation, or interface not available in perl: use c
- \* maps & graphs: use tcl/tk

- \* sophisticated graphics and higher math functions: use matlab
- \* web pages: might like php (perl also a choice)
- \* know/prefer some other language: it's up to you.

# Getting Started

- \* Learn basics
  - \* time
  - \* parameter files
  - \* databases
- \* Datascope Tutorial
- \* Software Development Tutorial
- \* Reference Guide



# Try something.

- \* something simple
- \* look at examples and modify
  - \* `$ANTELOPE/examples`
  - \* `http://www.indiana.edu/~aug/contrib.html`

# Extremely simple example

- \* Put a message on the screen

% template

Standard Antelope templates:

Distfile	perl	xpls
MAKE	ptk	xppl
Makefile	sh	xtcl
cs	simple.c	xvish
dbsimple.c	trsimple.c	xwish
manpage	<u>web.html</u>	xwish8
mx	xcdpl	
orbsimple.c	xpl	

```
% template xpl > note  
% template ptk >> note  
% chmod +x note  
% vile note  
... a little editing
```

```

: # use perl
eval 'exec $ANTELOPE/bin/perl -S $0 "$@"'
if 0;

use lib "$ENV{ANTELOPE}/data/perl" ;

use Tk ;
use ptkform ;

$MW = MainWindow->new ;

my @specs = <DATA> ;
ptkform($MW, \%var, \%widget, @specs) ;

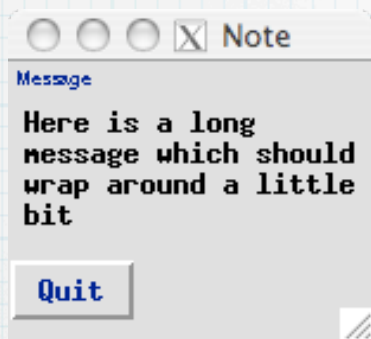
$widget{Quit}->configure(-command=>sub {$MW->destroy}) ;
$var{msg} = "@ARGV" ;

MainLoop ;

__DATA__
message msg      - 0,0 Message {message from the command line }
button  Quit     - +,0 Quit {I've seen enough }

```

```
% note Here is a long message which should wrap around a little bit
```



# Further reading

- \* Programming Perl
- \* Perl/Tk Pocket Reference
- \* perl-widget
- \* man ptkform
- \* man bookshelf
- \* [\\$ANTELOPE/antelope.html](#) / other man pages