Tcl-IOC

A Tcl shell for EPICS IOCs

or: Adding iocCore to Tcl

Dirk Zimoch, EPICS Meeting October 2005

What’s wrong with iocsh?

- iocsh is a very rudimentary shell
  - No programming capabilities
  - No loops
  - No branching
  - No user defined variables
  - Limited access to environment variables

➢ iocsh supports even less features than vxWorks shell

✓ Tcl has all these features

✓ Tcl syntax is similar to iocsh syntax

How are iocsh features ported to Tcl?

- Access to EPICS C-functions
  - A Tcl command is created for each registered EPICS function
  - All Tcl command arguments are translated to iocsh format
  - I/O redirection is handled
  - EPICS iocsh C-function is called
- Access to EPICS C-variables
  - A global Tcl variable is created for each registered EPICS variable
  - Read and write access to Tcl variable is redirected to EPICS variable

Which redirections are implemented?

- Standard redirections to files

  < filename, > filename, >> filename, *n*> filename, *n*>> filename

- Redirections to Tcl channels (open files, pipes, sockets)

  >@ channel, *n*>@ channel

- Redirections to Tcl variables

  >$ variable, >>$ variable, *n*>$ variable, *n*>>$ variable

- Redirection to result

  >$$, *n*>$$

How to use Tcl-IOC?

- Build tclioc library
  - tclioc compiles to a loadable library suitable for Tcl
  - tclioc is linked to standard EPICS libraries
  - tclioc contains standard iocsh commands
- Link application with tclioc instead of $(EPICS_BASE_IOC_LIBS)
- Build application as LOADABLE_LIBRARY instead of PROD
- Write Tcl startup script that loads application library before setting up IOC

Example Tcl-IOC Application

Makefile

```
TOP = ../..

include $(TOP)/configure/CONFIG

APP = myTclIocApp

#PROD = $(APP)   (replaced)
LOADABLE_LIBRARY = $(APP)

DBD += $(APP).dbd

$(APP)_SRCS += $(APP)_registerRecordDeviceDriver.cpp

#$(APP)_LIBS += $(EPICS_BASE_IOC_LIBS)   (replaced)
$(APP)_LIBS += tclioc

$(APP)_SRCS += myDriver.c

include $(TOP)/configure/RULES
```

Example Tcl-IOC startup script

```
#!/usr/bin/tcl

set app myTclIocApp

#load IOC into Tcl shell
load lib${app}.so tclioc
dbLoadDatabase ${app}.dbd
${app}_registerRecordDeviceDriver

#create records using TCL programming
for {set n 1} {$n <= 100} {incr n} {
       dbloadRecord example.template NUMBER=$n }

iocInit
dbl > records.txt

#go interactive
commandloop -prompt {puts -nonewline "epics> "}
```

What is required?

- Tcl (sources or development RPM)
  - http://tcl.sourceforge.net/
- recommended: Extended Tcl (TclX)
  - http://tclx.sourceforge.net/
- EPICS R3.14.7
  - http://www.aps.anl.gov/epics/base/R3-14/7.php
- Tcl-IOC sources
  - http://epics.web.psi.ch/software/tclioc/
- Linux (I have not yet tried other operating systems)