

**NAME**

snobol4tcl – SNOBOL4 Tcl/Tk interface

**SYNOPSIS**

**-INCLUDE 'stcl.sno'**

**telhandle = STCL\_CREATEINTERP()**

**STCL\_DELETEINTERP(telhandle)**

**STCL\_EVALFILE(telhandle,tclfilename)**

**value = STCL\_GETVAR(telhandle,varname)**

**STCL\_SETVAR(telhandle,varname,value)**

**STCL\_EVAL(telhandle,tclstmt)**

**DESCRIPTION**

Tcl is an embedable scripting language developed by John Osterhout, while at the University of California, Berkeley. Tk is a graphical user interface toolkit developed for Tcl.

This page describes STCL, an experimental facility for invoking Tcl and Tk from SNOBOL4 programs, inspired by Arjen Markus' "ftcl" FORTRAN/Tcl interface

**STCL\_CREATEINTERP** creates a Tcl interpreter and returns a handle which can be passed to the remaining functions.

**STCL\_DELETEINTERP** destroys a Tcl interpreter.

**STCL\_EVALFILE** reads a Tcl script file into the referenced Tcl interpreter.

**STCL\_GETVAR** retrieves the string value of named variable from a Tcl interpreter. **STCL\_GETVAR** stores a string value of named variable in a Tcl interpreter.

**STCL\_EVAL** evaluates a string containing Tcl code in a Tcl interpreter.

**EXAMPLE**

```

-INCLUDE 'stcl.sno'
  INTERP = STCL_CREATEINTERP()
  TCL_VERSION = STCL_GETVAR(INTERP, "tcl_version")
  OUTPUT = IDENT(TCL_VERSION) "Could not get tcl_version" :S(END)
  OUTPUT = "Tcl Version: " TCL_VERSION

* check Tcl version
  NUM = SPAN('0123456789')
  VPAT = NUM '.' NUM
  TCL_VERSION VPAT . VER          :S(CHECKV)
  OUTPUT = "could not parse tcl_version" :(END)

CHECKV LT(VER, 8.4)              :S(CHECKTK)

* Tcl 8.4 and later can dynamically load Tk!
  STCL_EVAL(INTERP, "package require Tk") :F(END)

* Check for Tk
CHECKTK TK_VERSION = STCL_GETVAR(INTERP, "tk_version") :F(NOTK)
DIFFER(TK_VERSION)          :S(HAVETK)

NOTK  OUTPUT = "Could not find tk_version" :(END)

HAVETK OUTPUT = "Tk version: " TK_VERSION

  STCL_EVAL(INTERP,
+   'button .hello -text "Hello, world" -command {set val hello};'
+   "pack .hello;"
+   'button .other -text "Other Choice" -command {set val other};'
+   "pack .other;"
+   "global val;"
+   "vwait val")

  OUTPUT = STCL_GETVAR(INTERP, "val")
END

```

**SEE ALSO**

tclsh(n), Tcl(n).

**AUTHOR**

Philip L. Budne

**BUGS**

NOTE! By default the STCL extension is not built into **snoBol4**(1), it must be explicitly included at build time. In Tcl 8.4 and later, Tk can be dynamically loaded by Tcl at runtime, but in earlier releases, it has to be included at compile time. When dynamically linked libraries are not available, this can cause the SNOBOL4 interpreter executable to expand by up to four fold! STCL should be a dynamically loaded SNOBOL4 extension.