

append *varName value ?value value ...?*

Append to variable

array *option arrayName ?arg arg ...?*

Manipulates array variables

array anymore *arrayName searchId*

array donesearch *arrayName searchId*

array names *arrayName*

array nextelement *arrayName searchId*

array size *arrayName*

array startsearch *arrayName*

break

Abort looping command

case *string ?in? patList body ?patList body ...?*

case *string ?in? {patList body ?patList body ...?}*

Evaluate one of several scripts, depending on a given value

This command is obsolete...

catch *script ?varName?*

Evaluate script and trap exceptional returns

cd *?dirName?*

Change working directory

close *fileId*

Close an open file

concat *?arg arg ...?*

Join lists together

continue

Skip to the next iteration of a loop

eof *fileId*

Check for end-of-file condition on open file

error *message ?info? ?code?*

Generate an error

eval *arg ?arg?*

Evaluate a tcl script

exec *?switches? arg ?arg ...?*

Invoke subprocess(es)

exit *?returnCode?*

End the application

expr *arg ?arg arg ...?*

Evaluate an expression

file *option name ?arg arg ...?*

Manipulate file names and attributes

file atime *name*
file dirname *name*
file executable *name*
file exists *name*
file extension *name*
file isdirectory *name*
file isfile *name*
file lstat *name varName*
file mtime *name*
file owned *name*
file readable *name*
file readlink *name*
file rootname *name*
file size *name*
file stat *name varName*
file tail *name*
file type *name*
file writable *name*

flush *fileId*

Flush buffered output for a file

for *start test next body*

"For" loop

foreach *varname list body*

Iterate over all elements in a list

format *formatString ?arg arg ...?*

Format a string in the style of sprintf

gets *fileId ?varName?*

Read a line from a file

glob *?switches? pattern ?pattern ...?*

Returns names of files that match patterns

global *varname ?varname ...?*

Access global variables

history *?option? ?arg arg ...?*
Manipulate the history list

if *expr1 ?then? body1 elseif expr2 ?then? body2 elseif ... ?else? ?bodyN?*
Execute scripts conditionally

incr *varName ?increment?*
Increment the value of a variable

info *option ?arg arg ...?*
Return information about the state of the Tcl interpreter

info args *procname*

info body *procname*

info cmdcount

info commands *?pattern?*

info complete *command*

info default *procname arg varname*

info exists *varName*

info global *?pattern?*

info level *?number?*

info library

info locals *?pattern?*

info patchlevel

info procs *?pattern?*

info script

info tclversion

info vars *?pattern?*

join *list ?joinString?*

Create a string by joining together list elements

lappend *varName value ?value value ...?*

Append list elements onto a variable

lindex *list index*

Retrieve an element from a list

insert *list index element ?element element ...?*

Insert elements into a list

list *?arg arg ...?*

Create a list

length *list*

Count the number of elements in a list

range *list first last*

Return one or more adjacent elements from a list

replace *list first last ?element element ...?*

Replace elements in a list with new elements

search *?mode? list pattern*

See if a list contains a particular element

lsort *?switches? list*

Sort the elements of a list

open *fileName ?access? ?permissions?*

Open a file

pid *?fileId?*

Retrieve process id(s)

proc *name args body*

Create a tcl procedure

puts *?-nonewline? ?fileId? string*

Write to a file

pwd

Returns the current working directory

read *?-newline?* *fileId*

read *fileId numBytes*

Read from a file

regexp *?switches?* *exp string ?matchVar? ?subMatchVar subMatchVar ...?*

Match a regular expression against a string

regsub *?switches?* *exp string subSpec varName*

Perform substitutions based on regular expression pattern matching

rename *oldName newName*

Rename or delete a command

return *?-code code? ?-errorinfo info? ?-errorcode code? ?string?*

Return from a procedure

scan *string format varName ?varName ...?*

Parse a string using conversion specifiers in the style of sscanf

seek *fileId offset ?origin?*

Change the access position for an open file

set *varName ?value?*

Read and write variables

source *fileName*

Evaluate a file as a Tcl script

split *string ?splitChars?*

Split a string into a proper Tcl list

string *option arg ?arg ...?*

Manipulate strings

string compare *string1 string2*

string first *string1 string2*

string index *string charIndex*

string last *string1 string2*

string length *string*
string match *pattern string*
string range *string first last*
string tolower *string*
string toupper *string*
string trim *string ?chars?*
string trimleft *string ?chars?*
string trimright *string ?chars?*

switch *?options? string pattern body ?pattern body ...?*

switch *?options? string {pattern body ?pattern body ...?}*

Evaluate one of several scripts, depending on a given value

tell *fileId*

Return current access position for an open file.

NOTE: this command is overridden in WavesWorld; use **tclTell** instead

time *script ?count?*

Time the execution of a script

trace *option ?arg arg ...?*

Monitor variable accesses

unknown *cmdName ?arg arg ...?*

Handle attempts to use non-existent commands

unset *name ?name name ...?*

Delete variables

uplevel *?level? arg ?arg ...?*

Execute a script in a different stack frame

upvar *?level? otherVar myVar ?otherVar myVar ...?*

Create link to variable in a different stack frame

while *test body*

Execute script repeatedly as long as a condition is met