NAME

Tcl_SplitList, Tcl_Merge, Tcl_ScanElement, Tcl_ConvertElement − manipulate Tcl lists

SYNOPSIS

#include <tcl.h>

int
Tcl_SplitList(*interp, list, argcPtr, argvPtr*)

char *
Tcl_Merge(*argc, argv*)

int
Tcl_ScanElement(*src, flagsPtr*)

int
Tcl_ConvertElement(*src, dst, flags*)

ARGUMENTS

Tcl_Interp	**interp*	(out)	Interpreter to use for error reporting.
char	**list*	(in)	Pointer to a string with proper list structure.
int	**argcPtr*	(out)	Filled in with number of elements in *list*.
char	****argvPtr*	(out)	**argvPtr* will be filled in with the address of an array of pointers to the strings that are the extracted elements of *list*. There will be **argcPtr* valid entries in the array, followed by a NULL entry.
int	*argc*	(in)	Number of elements in *argv*.
char	***argv*	(in)	Array of strings to merge together into a single list. Each string will become a separate element of the list.
char	**src*	(in)	String that is to become an element of a list.
int	**flagsPtr*	(in)	Pointer to word to fill in with information about *src*. The value of **flagsPtr* must be passed to **Tcl_ConvertElement**.
char	**dst*	(in)	Place to copy converted list element. Must contain enough characters to hold converted string.
int	*flags*	(in)	Information about *src*. Must be value returned by previous call to **Tcl_ScanElement**, possibly OR-ed with **TCL_DONT_USE_BRACES**.

DESCRIPTION

These procedures may be used to disassemble and reassemble Tcl lists. **Tcl_SplitList** breaks a list up into its constituent elements, returning an array of pointers to the elements using *argcPtr* and *argvPtr*. While extracting the arguments, **Tcl_SplitList** obeys the usual rules for backslash substitutions and braces. The area of memory pointed to by **argvPtr* is dynamically allocated; in addition to the array of pointers, it also holds copies of all the list elements. It is the caller's responsibility to free up all of this storage by calling

> **free**((char *) **argvPtr*)

when the list elements are no longer needed.

Tcl_SplitList normally returns **TCL_OK**, which means the list was successfully parsed. If there was a syntax error in *list*, then **TCL_ERROR** is returned and *interp->result* will point to an error message describing the problem. If **TCL_ERROR** is returned then no memory is allocated and **argvPtr* is not modified.

Tcl_Merge is the inverse of **Tcl_SplitList**: it takes a collection of strings given by *argc* and *argv* and generates a result string that has proper list structure. This means that commands like **index** may be used to extract the original elements again. In addition, if the result of **Tcl_Merge** is passed to **Tcl_Eval**, it will be parsed into *argc* words whose values will be the same as the *argv* strings passed to **Tcl_Merge**. **Tcl_Merge** will modify the list elements with braces and/or backslashes in order to produce proper Tcl list structure. The result string is dynamically allocated using **malloc()**; the caller must eventually release the space using **free()**.

If the result of **Tcl_Merge** is passed to **Tcl_SplitList**, the elements returned by **Tcl_SplitList** will be identical to those passed into **Tcl_Merge**. However, the converse is not true: if **Tcl_SplitList** is passed a given string, and the resulting *argc* and *argv* are passed to **Tcl_Merge**, the resulting string may not be the same as the original string passed to **Tcl_SplitList**. This is because **Tcl_Merge** may use backslashes and braces differently than the original string.

Tcl_ScanElement and **Tcl_ConvertElement** are the procedures that do all of the real work of **Tcl_Merge**. **Tcl_ScanElement** scans its *src* argument and determines how to use backslashes and braces when converting it to a list element. It returns an overestimate of the number of characters required to represent *src* as a list element, and it stores information in **flagsPtr* that is needed by **Tcl_ConvertElement**.

Tcl_ConvertElement is a companion procedure to **Tcl_ScanElement**. It does the actual work of converting a string to a list element. Its *flags* argument must be the same as the value returned by **Tcl_ScanElement**. **Tcl_ConvertElement** writes a proper list element to memory starting at **dst* and returns a count of the total number of characters written, which will be no more than the result returned by **Tcl_ScanElement**. **Tcl_ConvertElement** writes out only the actual list element without any leading or trailing spaces: it is up to the caller to include spaces between adjacent list elements.

Tcl_ConvertElement uses one of two different approaches to handle the special characters in *src*. Wherever possible, it handles special characters by surrounding the string with braces. This produces clean-looking output, but can't be used in some situations, such as when *src* contains unmatched braces. In these situations, **Tcl_ConvertElement** handles special characters by generating backslash sequences for them. The caller may insist on the second approach by OR-ing the flag value returned by **Tcl_ScanElement** with **TCL_DONT_USE_BRACES**. Although this will produce an uglier result, it is useful in some special situations, such as when **Tcl_ConvertElement** is being used to generate a portion of an argument for a Tcl command. In this case, surrounding *src* with curly braces would cause the command not to be parsed correctly.

KEYWORDS

backslash, convert, element, list, merge, split, strings