

**NAME**

Tcl\_SetResult, Tcl\_AppendResult, Tcl\_AppendElement, Tcl\_ResetResult – manipulate Tcl result string

**SYNOPSIS**

```
#include <tcl.h>
```

```
Tcl_SetResult(interp, string, freeProc)
```

```
Tcl_AppendResult(interp, string, string, ... , (char *) NULL)
```

```
Tcl_AppendElement(interp, string, noSep)
```

```
Tcl_ResetResult(interp)
```

```
Tcl_FreeResult(interp)
```

**ARGUMENTS**

Tcl_Interp	<i>*interp</i>	(out)	Interpreter whose result is to be modified.
char	<i>*string</i>	(in)	String value to become result for <i>interp</i> or to be appended to existing result.
Tcl_FreeProc	<i>freeProc</i>	(in)	Address of procedure to call to release storage at <i>string</i> , or <b>TCL_STATIC</b> , <b>TCL_DYNAMIC</b> , or <b>TCL_VOLATILE</b> .
int	<i>noSep</i>	(in)	If non-zero then don't output a space character before this element, even if the element isn't the first thing in the result string.

**DESCRIPTION**

The procedures described here are utilities for setting the result/error string in a Tcl interpreter.

**Tcl\_SetResult** arranges for *string* to be the return string for the current Tcl command in *interp*, replacing any existing result. If *freeProc* is **TCL\_STATIC** it means that *string* refers to an area of static storage that is guaranteed not to be modified until at least the next call to **Tcl\_Eval**. If *freeProc* is **TCL\_DYNAMIC** it means that *string* was allocated with a call to **malloc()** and is now the property of the Tcl system. **Tcl\_SetResult** will arrange for the string's storage to be released by calling **free()** when it is no longer needed. If *freeProc* is **TCL\_VOLATILE** it means that *string* points to an area of memory that is likely to be overwritten when **Tcl\_SetResult** returns (e.g. it points to something in a stack frame). In this case **Tcl\_SetResult** will make a copy of the string in dynamically allocated storage and arrange for the copy to be the return string for the current Tcl command.

If *freeProc* isn't one of the values **TCL\_STATIC**, **TCL\_DYNAMIC**, and **TCL\_VOLATILE**, then it is the address of a procedure that Tcl should call to free the string. This allows applications to use non-standard storage allocators. When Tcl no longer needs the storage for the string, it will call *freeProc*. *FreeProc* should have arguments and result that match the type **Tcl\_FreeProc**:

```
typedef void Tcl_FreeProc(char *blockPtr);
```

When *freeProc* is called, its *blockPtr* will be set to the value of *string* passed to **Tcl\_SetResult**.

If *string* is **NULL**, then *freeProc* is ignored and **Tcl\_SetResult** re-initializes *interp*'s result to point to the pre-allocated result area, with an empty string in the result area.

If **Tcl\_SetResult** is called at a time when *interp* holds a result, **Tcl\_SetResult** does whatever is necessary to dispose of the old result (see the **Tcl\_Interp** manual entry for details on this).

**Tcl\_AppendResult** makes it easy to build up Tcl results in pieces. It takes each of its *string* arguments and

appends them in order to the current result associated with *interp*. If the result is in its initialized empty state (e.g. a command procedure was just invoked or **Tcl\_ResetResult** was just called), then **Tcl\_AppendResult** sets the result to the concatenation of its *string* arguments. **Tcl\_AppendResult** may be called repeatedly as additional pieces of the result are produced. **Tcl\_AppendResult** takes care of all the storage management issues associated with managing *interp*'s result, such as allocating a larger result area if necessary. Any number of *string* arguments may be passed in a single call; the last argument in the list must be a NULL pointer.

**Tcl\_AppendElement** is similar to **Tcl\_AppendResult** in that it allows results to be built up in pieces. However, **Tcl\_AppendElement** takes only a single *string* argument and it appends that argument to the current result as a proper Tcl list element. **Tcl\_AppendElement** adds backslashes or braces if necessary to ensure that *interp*'s result can be parsed as a list and that *string* will be extracted as a single element. Under normal conditions, **Tcl\_AppendElement** will add a space character to *interp*'s result just before adding the new list element, so that the list elements in the result are properly separated. However, if *interp*'s result is empty when **Tcl\_AppendElement** is called, or if the *noSep* argument is 1, then no space is added.

**Tcl\_ResetResult** clears the result for *interp*, freeing the memory associated with it if the current result was dynamically allocated. It leaves the result in its normal initialized state with *interp->result* pointing to a static buffer containing **TCL\_RESULT\_SIZE** characters, of which the first character is zero. **Tcl\_ResetResult** also clears the error state managed by **Tcl\_AddErrorInfo** and **Tcl\_SetErrorCode**.

**Tcl\_FreeResult** is a macro that performs part of the work of **Tcl\_ResetResult**. It frees up the memory associated with *interp*'s result and sets *interp->freeProc* to zero, but it doesn't change *interp->result* or clear error state. **Tcl\_FreeResult** is most commonly used when a procedure is about to replace one result value with another.

## SEE ALSO

Tcl\_AddErrorInfo, Tcl\_SetErrorCode, Tcl\_Interp

## KEYWORDS

append, command, element, list, result, return value, interpreter