

NAME

Tcl_ExprLong, Tcl_ExprDouble, Tcl_ExprBool, Tcl_ExprString – evaluate an expression

SYNOPSIS

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

```
int
```

```
Tcl_ExprLong(interp, string, longPtr)
```

```
int
```

```
Tcl_ExprDouble(interp, string, doublePtr)
```

```
int
```

```
Tcl_ExprBoolean(interp, string, booleanPtr)
```

```
int
```

```
Tcl_ExprString(interp, string)
```

ARGUMENTS

Tcl_Interp	*interp	(in)	Interpreter in whose context to evaluate <i>string</i> .
char	*string	(in)	Expression to be evaluated. Must be in writable memory (the expression parser makes temporary modifications to the string during parsing, which it undoes before returning).
long	*longPtr	(out)	Pointer to location in which to store the integer value of the expression.
int	*doublePtr	(out)	Pointer to location in which to store the floating-point value of the expression.
int	*booleanPtr	(out)	Pointer to location in which to store the 0/1 boolean value of the expression.

DESCRIPTION

These four procedures all evaluate a string expression, returning the result in one of four different forms. The expression is given by the *string* argument, and it can have any of the forms accepted by the **expr** command. The *interp* argument refers to an interpreter used to evaluate the expression (e.g. for variables and nested Tcl commands) and to return error information. *Interp->result* is assumed to be initialized in the standard fashion when any of the procedures are invoked.

For all of these procedures the return value is a standard Tcl result: **TCL_OK** means the expression was successfully evaluated, and **TCL_ERROR** means that an error occurred while evaluating the expression. If **TCL_ERROR** is returned then *interp->result* will hold a message describing the error. If an error occurs while executing a Tcl command embedded in *string*, then that error will be returned.

If the expression is successfully evaluated, then its value will be returned in one of four forms, depending on which procedure is invoked. **Tcl_ExprLong** stores an integer value at **longPtr*. If the expression's actual value was a floating-point number, then it is truncated to an integer. If the expression's actual value was a non-numeric string then an error is returned.

Tcl_ExprDouble stores a floating-point value at **doublePtr*. If the expression's actual value was an integer, it is converted to floating-point. If the expression's actual value was a non-numeric string then an error is returned.

Tcl_ExprBoolean stores a 0/1 integer value at **booleanPtr*. If the expression's actual value was an integer or floating-point number, then **Tcl_ExprBoolean** stores 0 at **booleanPtr* if the value was zero and 1

otherwise. If the expression's actual value was a non-numeric string then an error is returned.

Tcl_ExprString returns the value of the expression as a string stored in *interp->result*. If the expression's actual value was an integer or floating-point number, then **Tcl_ExprString** converts it to string (using **sprintf** with a “%d” or “%g” converter).

KEYWORDS

boolean, double, evaluate, expression, integer, string