

NAME

Tcl_CreateCmdBuf, Tcl_AssembleCmd, Tcl_DeleteCmdBuf – buffer pieces of Tcl commands

SYNOPSIS

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

```
Tcl_CmdBuf
```

```
Tcl_CreateCmdBuf()
```

```
Tcl_DeleteCmdBuf(buffer)
```

```
char *
```

```
Tcl_AssembleCmd(buffer, string)
```

```
int
```

```
Tcl_CommandComplete(cmd)
```

ARGUMENTS

Tcl_CmdBuf	<i>buffer</i>	(in)	Token for a command buffer (the result of some previous call to Tcl_CreateCmdBuf).
char	<i>*string</i>	(in)	Additional piece of command input to be added to anything currently buffered.
char	<i>*cmd</i>	(in)	Command string to test for completeness.

DESCRIPTION

These procedures provide a convenient mechanism for assembling Tcl commands from an input source where command boundaries are not obvious. For example, if input is being read from a terminal, a user may type commands that span multiple lines. In situations like this, **Tcl_AssembleCmd** can be called with the individual lines as they are received. It buffers the lines internally and returns full commands when they are complete.

A command buffer is created by calling **Tcl_CreateCmdBuf**, and it is deleted by calling **Tcl_DeleteCmdBuf**. There may be any number of command buffers for a particular program or even for a particular interpreter; in most cases there should be one buffer for each independent source of command input.

When input arrives from a source you should call **Tcl_AssembleCmd**, passing it the new input as the *string* argument. **Tcl_AssembleCmd** will add the new input to anything currently buffered in *buffer*. If the information now buffered represents a complete Tcl command (i.e. there are no unclosed quotes, braces, brackets, or variable references), then **Tcl_AssembleCmd** returns a pointer to the complete command and arranges for the buffer to be cleared on the next call to **Tcl_AssembleCmd**. If the command is still incomplete (because, for example, there are unmatched braces) then **Tcl_AssembleCmd** returns NULL. **Tcl_AssembleCmd** keeps a private copy of the command being assembled, so that the caller need not preserve the contents of *string* between calls to **Tcl_AssembleCmd**. **Tcl_AssembleCmd** supports commands of arbitrary length (up to the total memory limit imposed by the operating system, if any).

The procedure **Tcl_CommandComplete** takes a Tcl command string as argument and determines whether the command string is complete in the sense defined above. If so then it returns 1; otherwise it returns 0.

KEYWORDS

assemble, buffer, partial command