

NAME

bgerror – Command invoked to process background errors

SYNOPSIS

bgerror *message*

DESCRIPTION

Release 8.5 of Tcl supports the **interp bgerror** command, which allows applications to register in an interpreter the command that will handle background errors in that interpreter. In older releases of Tcl, this level of control was not available, and applications could control the handling of background errors only by creating a command with the particular command name **bgerror** in the global namespace of an interpreter. The following documentation describes the interface requirements of the **bgerror** command an application might define to retain compatibility with pre-8.5 releases of Tcl. Applications intending to support only Tcl releases 8.5 and later should simply make use of **interp bgerror**.

The **bgerror** command does not exist as built-in part of Tcl. Instead, individual applications or users can define a **bgerror** command (e.g. as a Tcl procedure) if they wish to handle background errors.

A background error is one that occurs in an event handler or some other command that did not originate with the application. For example, if an error occurs while executing a command specified with the **after** command, then it is a background error. For a non-background error, the error can simply be returned up through nested Tcl command evaluations until it reaches the top-level code in the application; then the application can report the error in whatever way it wishes. When a background error occurs, the unwinding ends in the Tcl library and there is no obvious way for Tcl to report the error.

When Tcl detects a background error, it saves information about the error and invokes a handler command registered by **interp bgerror** later as an idle event handler. The default handler command in turn calls the **bgerror** command. Before invoking **bgerror**, Tcl restores the **errorInfo** and **errorCode** variables to their values at the time the error occurred, then it invokes **bgerror** with the error message as its only argument. Tcl assumes that the application has implemented the **bgerror** command, and that the command will report the error in a way that makes sense for the application. Tcl will ignore any result returned by the **bgerror** command as long as no error is generated.

If another Tcl error occurs within the **bgerror** command (for example, because no **bgerror** command has been defined) then Tcl reports the error itself by writing a message to stderr.

If several background errors accumulate before **bgerror** is invoked to process them, **bgerror** will be invoked once for each error, in the order they occurred. However, if **bgerror** returns with a break exception, then any remaining errors are skipped without calling **bgerror**.

If you are writing code that will be used by others as part of a package or other kind of library, consider avoiding **bgerror**. The reason for this is that the application programmer may also want to define a **bgerror**, or use other code that does and thus will have trouble integrating your code.

EXAMPLE

This **bgerror** procedure appends errors to a file, with a timestamp.

```
proc bgerror {message} {
    set timestamp [clock format [clock seconds]]
    set fl [open mylog.txt {WRONLY CREAT APPEND}]
    puts $fl "$timestamp: bgerror in $::argv '$message'"
    close $fl
}
```

SEE ALSO

after(n), errorCode(n), errorInfo(n), interp(n)

KEYWORDS

background error, reporting