

**NAME**

option – Add/retrieve window options to/from the option database

**SYNOPSIS**

**option add** *pattern value ?priority?*

**option clear**

**option get** *window name class*

**option readfile** *fileName ?priority?*

**DESCRIPTION**

The **option** command allows you to add entries to the Tk option database or to retrieve options from the database. The **add** form of the command adds a new option to the database. *Pattern* contains the option being specified, and consists of names and/or classes separated by asterisks or dots, in the usual X format. *Value* contains a text string to associate with *pattern*; this is the value that will be returned in calls to **Tk\_GetOption** or by invocations of the **option get** command. If *priority* is specified, it indicates the priority level for this option (see below for legal values); it defaults to **interactive**. This command always returns an empty string.

The **option clear** command clears the option database. Default options (from the **RESOURCE\_MANAGER** property or the **.Xdefaults** file) will be reloaded automatically the next time an option is added to the database or removed from it. This command always returns an empty string.

The **option get** command returns the value of the option specified for *window* under *name* and *class*. If several entries in the option database match *window*, *name*, and *class*, then the command returns whichever was created with highest *priority* level. If there are several matching entries at the same priority level, then it returns whichever entry was most recently entered into the option database. If there are no matching entries, then the empty string is returned.

The **readfile** form of the command reads *fileName*, which should have the standard format for an X resource database such as **.Xdefaults**, and adds all the options specified in that file to the option database. If *priority* is specified, it indicates the priority level at which to enter the options; *priority* defaults to **interactive**.

The *priority* arguments to the **option** command are normally specified symbolically using one of the following values:

**widgetDefault**

Level 20. Used for default values hard-coded into widgets.

**startupFile**

Level 40. Used for options specified in application-specific startup files.

**userDefault**

Level 60. Used for options specified in user-specific defaults files, such as **.Xdefaults**, resource databases loaded into the X server, or user-specific startup files.

**interactive**

Level 80. Used for options specified interactively after the application starts running. If *priority* is not specified, it defaults to this level.

Any of the above keywords may be abbreviated. In addition, priorities may be specified numerically using integers between 0 and 100, inclusive. The numeric form is probably a bad idea except for new priority levels other than the ones given above.

**EXAMPLES**

Instruct every button in the application to have red text on it unless explicitly overridden:

```
option add *button.foreground red startupFile
```

Allow users to control what happens in an entry widget when the Return key is pressed by specifying a script in the option database and add a default option for that which rings the bell:

```
entry .e
```

```
bind .e <Return> [option get .e returnCommand Command]
```



option(3tk)

Tk Built-In Commands

option(3tk)

**option add** \*.e.returnCommand bell widgetDefault**KEYWORDS**

database, option, priority, retrieve

