

Best way to set up a dict of defaults?

Source: <http://coding.derkeiler.com/Archive/Tcl/comp.lang.tcl/2007-09/msg00104.html>

- *From:* Twylite <twylite.crypt@xxxxxxxxxx>
 - *Date:* Tue, 04 Sep 2007 14:50:25 -0000
-

I have a preferences class that acts something like a persistent dict. One of parameters to initialise the class is a list of name/value pairs representing the configuration defaults.

But some of the default values are calculated/derived rather than static (say based on the name of the application, \$argv0). What is the best way for me to set up these defaults?

I current use list, which is ugly because of line continuation:

```
set app_path [file dirname $argv0]
set app_base_name [file tail [file rootname $argv0]]
set prefs_defaults [list \
log.file [file join $app_path "data" "${app_base_name}.log"] \
log.level WARN \
]
```

Subst appears to be a solution, but can trick the unwary:

```
set prefs_defaults [subst {
log.file [file join $app_path "data" "${app_base_name}.log"]
log.level WARN
}]
```

Notice that this will work correctly only so long as the resulting file path contains no spaces, otherwise you will get e.g.

```
log.file C:/Program Files/MyApp/data/myapp.log
which is 3 elements instead of 2 ;(
```

Am I missing something obvious; is there a better way?

Regards, Twylite

.