

---

Subject: Re: Objects with Widgets, Save/Restore

Posted by [John-David T. Smith](#) on Fri, 03 Aug 2001 20:30:35 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

David Fanning wrote:

>  
> I wrote a minute ago:  
>  
>> I played with it for days trying to keep the  
>> current GUI (which, of course, has the RESTORE  
>> button on it), but no matter what I tried (and  
>> I thought I tried some pretty sophisticated  
>> things) I couldn't get it to work. So now when  
>> you hit the RESTORE button, the GUI disappears  
>> and comes back a second later with the old  
>> session up and ready to go.  
>  
> After reading JD's post, I'm pretty sure  
> there is a way to get this to work. In fact,  
> I believed that for the 3-4 frustrating days  
> I spent working on the problem. I *\*still\**  
> believe it. I just can't get it to work. :-(  
>

The secret is this same notion of pruning out the bits you don't need.  
Here's a sketch of how I'd approach your problem.

```
pro DavidsPlayhouse::Restore
  oldself=self          ;for killing later
  wlInfo=self.wlInfo    ;save the GUI, eat soy products
  self.wlInfo=ptr_new() ;detach, avoiding the carnage
  restore_obj, self.SaveFile ;travel back in time
  obj_destroy,oldself    ;kill our old self, except wlInfo
  self.wlInfo=wlInfo     ;reattach our saved widget info
  self->UpdateGUI        ;I'm not who I think I am
end
```

where "restore\_obj" restores an object on top of itself. At least I think this is what you're trying to do. I once called this "transmogrification" ... to the delight and ridicule of various newsgroup regulars. It relies on the fact that the variable "self" is actually passed by reference to all methods, so you can overwrite it simply by restoring from file an object previously saved under a variable name "self" (try "self=1" in a method sometime and see what trouble you'll get yourself into). Hopefully, this is an object of the same type, or you'll be in for some surprises. Notice how I killed my old self (after carefully detaching the GUI components I wanted to escape the Cleanup carnage), to avoid having a split personality (and

memory leaks).

The secret to keeping the same GUI running across this restore process is simple: isolate all widget state info under a pointer, detach it beforehand, and retain it across the restore, for grafting onto the newly reanimated object. As long as all widget ID's and state information survive the (wait for it) transmogrification, you won't have any GUI flashing, since it'll be the same damn widgets running all throughout (yes, including the button which invoked the command). I'd also isolate the restore code in some error checking. I can send you an entire example if you're interested. And by the way, to avoid the confusion Pavel identified, I call this feature "Revert from Disk...", as in "Go back to that version I saved before I did that stupid thing". It's very handy. Do also consult the `restore_obj` writeup on your own website for some of the potential snafu's associated with object restoration.

Good luck,

JD

---