
Subject: Re: Catalyst Object Widget Hierarchy
Posted by [David Fanning](#) on Thu, 17 Dec 2009 01:40:20 GMT
[View Forum Message](#) <> [Reply to Message](#)

Jean-Paul Davis writes:

> Thought I'd keep this discussion here since my next question is so
> closely related: what is the purpose of the WidgetBase::ADD and
> WidgetAtom::ADD methods? When creating an object widget hierarchy
> using Catalyst, I see in your examples that you simply create the
> individual widget objects from within the top-level object's INIT or
> GUI method, using named variables for the object references only when
> needed as the parent argument to a child widget or as a property of
> the top-level object. Would there ever be any reason to "ADD" child
> widget objects to parent widget objects?

Thanks very much for your questions. It's nice to know
a year's worth (at least!) of unpaid effort is at least
interesting to someone else. :-)

To answer your question, I don't know exactly how the
WidgetBase::Add (or WidgetAtom::Add) method would be
useful. I haven't had occasion yet to use them, I don't
think. But we very deliberately over-engineered the
Catalyst framework to include functionality we couldn't
yet imagine or think we would need.

I believe I was thinking of any object in the system as
a sort of glorious widget "user value". That is, since
every object is a subclassed container widget, it can
hold as many "other" objects as you can imagine needing.

I can distinctly remember taking a shower when one
advantage of this way of thinking occurred to me. I
suddenly realized that one way of adding complexity
to images was to be able to put things on top of them
in "layers". I could have a text layer, a map grid layer,
a station data layer, and so forth. Each layer could
be turned on and off just by setting its "visibility"
property. About two hours later, with soap still in my
ears, I think, I had written the AnnotateWindow program,
which contained functionality I have never seen before
in a direct graphics program.

My point is, you just never know when something is going
to be useful. Dave Burrige has many faults (in case he
is reading this!), but one of his many strengths as a
programmer is not closing off possibilities too early

in the process. I hope I learned that from him.

> I know you've been asked before, but do you think there's even a
> remote chance that someone (you, Burridge, or even someone else) might
> ever write a book on how to use Catalyst?

Yes, I think there is a chance. I'd love to write the book myself. But trying to write a book, hold down a full-time job, play an occasional tennis match, and spend time with your family is a sure-fire recipe for marital disaster. Doing it one time nearly cost me my marriage. So far I haven't been willing to anger the gods by attempting it twice.

But who knows. Divorce (for any number of *other* reasons, God knows), retirement, a better job situation so that children who made it through college can actually leave home and quit being a sponge on their parent's finances, and even hitting the Lotto could all happen tomorrow. (Of course, I would have to buy a ticket.) If they did, I'd be at that book like a dog on a bone. Providing, of course, that someone would be willing to pay me for it.

My biggest hope is that YOU will become interested enough to write the book, Jean-Paul. ;-)

Cheers,

David

--

David Fanning, Ph.D.

Fanning Software Consulting, Inc.

Coyote's Guide to IDL Programming: <http://www.dfanning.com/>

Sepore ma de ni thui. ("Perhaps thou speakest truth.")
