Subject: Re: passing parameters from base to base
Posted by Pavel A. Romashkin on Mon, 02 Dec 2002 16:49:31 GMT

View Forum Message <> Reply to Message

Aha, I knew it! :-)
Yes, I think this is adequate. And what David is talking about seems
pretty much the same and Ok too, as long as traditional relationships
among programs are preserved, i.e. one is launched from another, or a
known/predictable sequence of launches exists. As long as one can/wants
to *pass* anything to another process, everything seems trivial, and
messaging system (as it appears to me) is a neat way of inter-object communication.
I faced this too but what I wanted is absolutely no assumptions
regarding what will launch when, and I did not want to keep any
notations of what *has* already launched. It should all happen by
itself. As long as one is allowed to have a Common variable, this is
simple to achieve because you have a place to keep your "watchdog"
object, but if you wanted to get by without Commons, it is more
difficult and is not nearly as neat anymore.
I experimented quite a bit with non-destructable (or, rather,
self-recreating if deleted by mistake) objects that maintained links to
themselves and others, sort of being a virus infecting all of my
programs. The object is tiny, so maintaining it was not taxing to the
system. This worked to some degree but I never finished it, had to
actually do some work.
Therefore, I agree that the proposed solutions will have to do :-)
Cheers,
Pavel
P.S. BTW, I think that "relations and introductions can be done at
compile time" is an overstatement. Objects have to exist in order to
link anything to them, because their location in heap memory is not
known until then.

JD Smith wrote:
>
> and oFunk could sign up for the tvDraw messages it needs. Perhaps
> oFunk even builds an ObjMsg object of its own and signs up with it
> directly. Usually this is perfectly adequate: relations and
> introductions can be done at compile time or Init time. This works
> well for, e.g., building a viewer shell with all the desired plug-ins,
> especially when you need to carry widget bases around to build the
> interface inside.
>

... snip ...

>
> Any suggestions much appreciated. Another option is to publish
> "capabilities" or lists of potential messages sent in a common block

> of some kind, but that gets complicated with many multiples of the
> same type of object on the global stack.
>
> JD