Subject: Re: IDL widgets: comments and questions
Posted by cannon on Fri, 17 Mar 1995 09:27:52 GMT
View Forum Message <> Reply to Message

In article <3k7396$6qi@hermod.uio.no>, steinhh@amon.uio.no (Stein Vidar Hagfors Haugan) writes:
|>

|> If you set it up correctly using a tree structure, you only need
|> one widget ID in order to find your information no matter how complex
|> your application is. The places where you have a legitimate
|> need for that ID is inside the event handler (or routines called by the
|> event handler). And in the event handler, you do have EVENT.TOP
|> (and even EVENT.HANDLER/EVENT.ID!). This, in combination with structure
|> variables as UVALUES, removes any need for common blocks in order
|> to pass data anywhere.
|>
.... cut ....

I'm trying to adopt your philosophy, but how do you handle a function which uses a widget to return a value (like the pickfile routine say, which uses a common block)? You call the function which creates the widgets and it is all fine until you want to exit, with the value you want the function to return in a uvalue somewhere. But to exit you have to destroy the widgets so control gets handed back to the function which made them - and then it cant get at the uvalue any longer to return the thing you want??

The only examples i've seen use common blocks, but is there any other way round it?

Robert Cannon