Subject: Re: IDL 5.0 observations
Posted by Stein Vidar Hagfors H on Thu, 05 Jun 1997 07:00:00 GMT
View Forum Message <> Reply to Message

Another change to the widget handling that I've found
is that calling XMANAGER without any arguments will no
longer resume event processing when called from "within"
another XMANAGER call.

Some times, I like to be able to have widget "A"'s event
procedure "A_EVENT" read in data, start off widget B
from within A_EVENT to display the data, *without* making
A modal (so you can read and display other data sets from A),
but still let the call to B *not* "fall through" - i.e., to
delay the return from B until B actually dies and no longer
needs the data passed to it.

This is handy for those situations where B uses a NO_COPY
operation that would make it's arguments undefined upon
an immediate return.

I'm aware that making B "hang" without modalizing A will
allow A to be killed before killing B (when the return from
B to A_EVENT comes, A may not be valid), but that's quite
OK: The point is that the arguments passed to B may have to
be returned to the user who started A from the prompt, so
A_EVENT needs to get it's hands on the arguments to B *after*
B dies.

In v 4.0.1 one could do this by using something like

```
XMANAGER,"B",BASE
WHILE WIDGET_INFO(BASE,/VALID_ID) DO XMANAGER
```

in program B, but this now causes an infinite loop calling
XMANAGER without causing events to be processed (but taking
100% cpu power).

This may not be a big problem, but it's at least a significant
change in the way XMANAGER behaves.

Stein Vidar