In article <3tv6bs$3e9@news2.ucsd.edu>, pierce@magik.ucsd.edu (Ben Stephens) writes:
> Even this already rather inelegant strategy did not suffice. IDL
> wouldn't let me access the UVAL stash of my dying widget from my CLEAN
> routine, so I had to leave a copy of the state in the common block just
> for its use.
>
> Basically, I'm frustrated because I never had to deal with any of this
> under the CW_LOADSTATE and CW_SAVESTATE system (Sure, bugs resulted from
> IDL's re-use of IDs, but now that IDL never re-uses IDs, that shouldn't
> be a problem). So that's why I'm asking
>
> Has anyone else come up with a more elegant solution to this
>   problem?
>

I have recently begun to use handles to get around some of these problems.
Rather than storing STATE in the UVAL, create a handle and just store the
HANDLE_ID in the UVAL. Handles allow the /NO_COPY behavior for efficiency, just
like widgets do. Using handles you should still be able to access STATE even
on a dying widget, as long as CLEANUP can determine HANDLE_ID. With handles it
is also possible to have multiple copies of the HANDLE_ID, which can ease
problems of routines communicating with each other. As with UVALs, only one
routine can "check out" the handle with /NO_COPY at once. If a routine needs
to call another routine which needs to access STATE then it must must either
"check it back in" or pass it directly.

_____

Mark Rivers   (312) 702-2279 (office)
CARS   (312) 702-9951 (secretary)
Univ. of Chicago   (312) 702-5454 (FAX)
5640 S. Ellis Ave.   (708) 922-0499 (home)
Chicago, IL 60637   rivers@cars3.uchicago.edu (Internet)