Subject: MGS_GUIObject -> MGS_BaseGUI
Posted by Martin Schultz on Mon, 09 Apr 2001 15:16:53 GMT
View Forum Message <> Reply to Message

Hi all,

JD will whine "now he's even taking over a complete thread with his initials ..." ;-)

Nevertheless: A couple of comments that I received are telling me that I am on the right way here. So, I decided to proceed and keep you informed about (at least some) steps along the way.

Version 1.1 of the GUI objects is now on
ftp://ftp.dkrz.de/pub/Outgoing/martin_schultz/idl

Changes:
(1) renamed base GUI object to MGS_BaseGUI (well, yes, errrrh... the "object" really wasn't necessary ;-) -- of course this involves several changes in all inheriting objects (Ctrl-s or Meta-% are very helpful here)

(2) improved layout capabilities for compound container: in the first version, all compound widgets were placed as children from the GUI object's layoutID widget. Now, you can give the compound widget a "hook number" so that it can be attached to any base that you define in the BuildGUI method. The BaseGUI object now has an extra pointer field called cwbaseid where you have to store the respective widget IDs when you build your widget. This is documented (by example) in the BaseGUI object, and to see a working example, check out the new MGS_MapSetup object.
-- If you don't mind having your compound widgets attached directly to the layoutID, no changes are necessary.

Next steps:
I guess I can no longer get around a more in-depth analysis of the event handling ;-(I'll probably do some reading in David's book tonight and then try to put it in correctly. What does not yet work as I think it should is the processing of field events:
(a) the field should accept "illegal" values until it looses it's focus (example: you start typing a negative number with "-" ; currently, you'll have to enter "1", then move back and insert the "-")... yet, illegal characters should be banned immediately (e.g. no letters in a number field, ...)
(b) when the field looses it's focus or when queried from elsewhere (UpdateObject?) it should notify a parent object if there is one. I am currently thinking of implementing a CW_Event_Handler in addition to

the generic Event_Handler method, which can then be overwritten to do
widget specific things (example: test if the lower bound value is
lower than the upper bound value in MGS_RangeField).
(c) I am currently unsure, which widget events need to be captured and
processed in order to make it work correctly under each and every turn
and twist a user (those non-standardized humans ;-) can think of.
Here, I admit that I lack years of experience - and I can only hope to
find this expertise in the famous book which I have not yet fully
memorized ;-(
Examples:
A kill event will now always leave the object itself untouched.
Should I call a special method that reacts like the Close or Cancel
buttons are hit? Do I need to do more than that?
I haven't worried about resizing at all, yet. Sure, this would be a
nice feature for things like Ben's twolist object.

If there is anyone out there who would like to give me a few hints
concerning these issues, I would greatly appreciate your valuable
comments. But - then again - I don't really expect you to comment,
because, after all, I am developing MGS_BaseGUI to free you (and
myself ;-) from exactly those details ...

Cheers,

Martin

--
[[[[[[[[[
[[Dr. Martin Schultz Max-Planck-Institut fuer Meteorologie [[
[[Bundesstr. 55, 20146 Hamburg [[
[[phone: +49 40 41173-308 [[
[[fax: +49 40 41173-298 [[
[[martin.schultz@dkrz.de [[
[[[[[[[[[[