

---

Subject: Re: passing parameters from base to base  
Posted by [Pavel A. Romashkin](#) on Thu, 21 Nov 2002 20:52:46 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

The simplest way is to pass a few parameters to SetParams that reside in the Mainbase STATE structure. I personally, for simplicity sake, pass the entire STATE structure (which gets passed by reference and is easy to update in that case).

Of course, it is cleaner if you only pass fields of STATE (that are pointers because otherwise you'd be passing state.fields by value but if you alter \*state.fields, these will get updated).

If you want to update, say, labels on Mainbase, you can simply call WIDGET\_CONTROL on the State.MainBaseID inside SetParams.

Or, if you want to really do something intricate with the results returned by SetParams, you could do something like depicted on [http://www.ainaco.com/idl/idl\\_library/export\\_to\\_main.pro](http://www.ainaco.com/idl/idl_library/export_to_main.pro) which will interrupt the main event handler until dismissed.

Cheers,  
Pavel

Gert wrote:

>  
> Hi,  
>  
> I've been trying to figure this one out for a while. I have 2 bases. If in  
> Mainbase the button Set is pushed, a second base SetParams is called. Stuff  
> happens there and the idea is that if SetParams is killed, a series of  
> numbers go back to Mainbase. Now how can you write this neatly, so that the  
> code for the SetParams can easily be used in other progs?  
> These are my thoughts:  
> I could pass a pointer to SetParams that keeps the desired data, but how  
> does Mainbase know that  
> SetParams is killed and that it needs to update its fields?  
> I looked at the examples for compound widgets (e.g. cw\_defroi) and these do  
> the trick but they do not use xmanager. Is this the only way.  
>  
> thx for any help,  
>  
> Gert

---