Subject: Re: Returning Values in iTools
Posted by David Alexander on Tue, 10 Jan 2006 23:31:31 GMT
View Forum Message <> Reply to Message

James,

Yeah, I guess iTools weren't designed to be run in a blocking-type paradigm.

One idea is to create an object that calls either your wrapper routine, or, if you don't have one, that calls IDLITSYS_CREATETOOL directly. Then implement a callback method on that object that gets called when the user closes the tool.

If the user clicks on your custom button, you can handle the call to your callback easily. The tricky part is: what happens when the user selects the File->Exit menu item, or even just clicks on the X button in the toolbar?

This easiest way is if you have implemented your own tool widget code (ie, code that replaces IDLitWdTool). In this case, you can catch the WIDGET_KILL_REQUEST event and call the callback method on your object.

Otherwise, you have to do a little more work:

I took a quick look at the IDL code, and in both the File->Exit case and the "user clicks X" case the itools' SHUTDOWN service is called. So, you could replace the SHUTDOWN service with your own object, which should subclass from the iTools shutdown service class. This is slightly hacky (since the shutdown service is not documented), but there's no good documented way to do this.

So you could do something like this:

1) Create a class that subclasses from IDLitOpShutdown (this is the itools shutdown service class).
2) Implement a DoAction method in this class that calls the parent DoAction method, then calls your object's callback.
3) After you call IDLITSYS_CREATETOOL, add your subclassed service that will replace the itools shutdown service:

```
   ;call IDLITSYS_CREATETOOL, then:
   void=itGetCurrent(TOOL=oTool)
   oService=OBJ_NEW('MyShutdownService',IDENTIFIER='SHUTDOWN')
   oTool->AddService,oService
```

You could also put the last two lines in the Init method of your tool class, which is probably a more approriate place. Using the IDENTIFIER

keyword when creating your service object is important, since that's
the identifier that the itools system uses to get the shutdown service
when the user quits the tool (See the handling of the
WIDGET_KILL_REQUEST event in IDLitwdTool_event for an example. This is
in IDLitWdTool.pro).

Dave

Subject: Re: Returning Values in iTools
Posted by David Alexander on Tue, 10 Jan 2006 23:42:39 GMT

View Forum Message <> Reply to Message

Hmm, well, looking more closely at the IDLitOpShutdown class, it's a
little more complicated.

When the user tries to exit the tool, if there are any changes, they
are prompted whether they want to save the session. If they click
'Cancel', the tool doesn't close.

So in that case, the DoAction method you implement in your special
shutdown service class should take that into account, something like
this:

```
function MyShutdownService::DoAction,oTool

   ;Need to get the object that called IDLITSYS_CREATETOOL here.
   ;You could store it somewhere in the tool object.

   ;Then, call superclass
   result=self->IDLitOpShutdown::DoAction(oTool)

   ;See if we're really shutting down.
   ;If we are, the superclass will have destroyed the tool object.
   if ~OBJ_VALID(oTool) then $
      ;Call the callback on the object that created
IDLITSYS_CREATETOOL

   return,OBJ_NEW()
end
```