Subject: Re: Properties
Posted by David Fanning on Fri, 12 Dec 2003 22:07:31 GMT
View Forum Message <> Reply to Message

JD Smith writes:

> OK, I'm trying to hijack the new Properties capabilities of iTools for
> my own devious purposes, and I've managed whip up this test:
>
> pro TestProps::Config
> b=widget_base(/COLUMN)
> ; Create and initialize the component.
> p=widget_propertysheet(b,VALUE=self)
> widget_control,b,SET_UVALUE=self,/REALIZE
> XManager,'TestPropsPropertySheet',b,/NO_BLOCK
> end
>
> function TestProps::Init
> if self->IDLitComponent::Init(NAME='TestProps') eq 0 then return,0
> self->RegisterProperty,'MY_FIRST_PROPERTY',/BOOLEAN, $
> NAME='Propertus Incipiens'
> return,1
> end
>
> pro TestProps__define
> st={TestProps, $
> INHERITS IDLitComponent, $
> data:0}
> end
>
> Sadly enough, here's what happens:
>
> IDL> a=obj_new('testprops')
> IDL> a->Config
> % Keyword MY_FIRST_PROPERTY not allowed in call to: WIDGET_PROPERTYSHEET
> % Execution halted at: TESTPROPS::CONFIG 5
> /home/jdsmith/idl/irs_cubism/cubism/cube/testprops__define.p ro
> % $MAIN$
>
> I'm really not sure how a property ID is getting transformed into a
> keyword in this call... sounds like some internal _EXTRA hi-jinx to
> me. What ingredient am I missing?

What you are missing are GetProperty and SetProperty methods
that allow use of the keyword MY_FIRST_PROPERTY. Add those,
and you will be in business! :-)

These are required because this is how IDLitComponent can

get and set values, as needed.

Cheers,

David
--
David W. Fanning, Ph.D.
Fanning Software Consulting, Inc.
Phone: 970-221-0438, E-mail: david@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: Properties
Posted by David Fanning on Fri, 12 Dec 2003 22:13:44 GMT
View Forum Message <> Reply to Message

David Fanning writes:

> What you are missing are GetProperty and SetProperty methods
> that allow use of the keyword MY_FIRST_PROPERTY. Add those,
> and you will be in business! :-)

Be sure, by the way, to call the IDLitComponent superclass
GetProperty and SetProperty methods from within your own!
This means the GetPropery method will probably need a
_REF_EXTRA keyword.

Sheesh! Doesn't this stuff get deep quickly! :-(

Like this:

```
 PRO TestProps, MY_FIRST_PROPERTY=my_first_property, _Ref_Extra=extra
   my_first_property = self.data
   self -> IDLitComponent::GetProperty, _Extra=extra
END
```

Cheers,

David
--
David W. Fanning, Ph.D.
Fanning Software Consulting, Inc.
Phone: 970-221-0438, E-mail: david@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: Properties
Posted by JD Smith on Sat, 13 Dec 2003 01:24:54 GMT
View Forum Message <> Reply to Message

On Fri, 12 Dec 2003 15:13:44 -0700, David Fanning wrote:

> David Fanning writes:
>
>> What you are missing are GetProperty and SetProperty methods that allow
>> use of the keyword MY_FIRST_PROPERTY. Add those, and you will be in
>> business! :-)
>
> Be sure, by the way, to call the IDLitComponent superclass GetProperty
> and SetProperty methods from within your own! This means the GetPropery
> method will probably need a _REF_EXTRA keyword.
>
> Sheesh! Doesn't this stuff get deep quickly! :-(
>
> Like this:
>
> PRO TestProps, MY_FIRST_PROPERTY=my_first_property, _Ref_Extra=extra
> my_first_property = self.data
> self -> IDLitComponent::GetProperty, _Extra=extra
> END
>
>

Thanks David. And, to my utter and complete surprise, you actually have a to implement the getting and setting of a property via the keyword in the (Get|Set)Property methods, you can't fool it by just putting the keyword there. I gather it must give a value a round trip through obj->Set,obj->Get to see whether your (Get|Set)Property methods actually do as advertised. Very sneaky, actually checking on you like that. Makes me feel like big brother is watching.

JD