
Subject: Re: IDLitComponent - Introspection

Posted by [marc schellens\[1\]](#) on Wed, 11 May 2005 12:18:30 GMT

[View Forum Message](#) <> [Reply to Message](#)

There is no way except what you already mentioned.

IDLitComponent::GetPropertyByIdentifier isn't written in IDL but is an internal routine.

Cheers,
marc

Subject: Re: IDLitComponent - Introspection

Posted by [David Fanning](#) on Wed, 11 May 2005 13:39:34 GMT

[View Forum Message](#) <> [Reply to Message](#)

Robert Barnett writes:

> I've been trying to write my own classes which incorporate IDLitComponent and widget_propertySheet. This allows the user to tinker with object properties via a GUI,
>
> I'm very confused by IDLitComponent::GetPropertyByIdentifier. The method allows you to do a "ClassName::GetProperty, Name=Value",
> however "Name" is only known at runtime.
>
> For example
>
> obj -> GetPropertyByIdentifier("BoringProperty", empty_variable)
>
> somehow triggers
>
> obj -> GetProperty, BORINGPROPERTY=empty_variable
>
> This is something that I would very much like to do with my own classes without having to inherit IDLitComponent.
>
> To be more precise, does anyone know how IDLitComponent::GetPropertyByIdentifier tricks the arg_present()
> function into thinking that a keyword needs to be set?

Here is a quick and dirty explanation of how this can be done.
The actual code is a *bit* more complicated than this, but not much more complicated. In any case, this will be enough to get you started.

The main trick is to define an output variable on the *__DEFINE procedure so you can get the class structure easily. All depends on that. :-)

```
.*****  
,
```

```
PRO Object_Trick::GetProperty, _Ref_Extra=extra
```

```
Object_Trick__Define, class  
properties = Tag_Names(class)  
keyword = (extra)[0]  
index = Where(properties EQ keyword, count)  
IF count GT 0 THEN BEGIN  
  (Scope_Varfetch(keyword, /Ref_Extra)) = self.(index)  
ENDIF
```

```
END
```

```
FUNCTION Object_Trick::INIT
```

```
self.thick = 2  
self.linestyle = 4
```

```
RETURN, 1
```

```
END
```

```
PRO Object_Trick__Define, class
```

```
class = { Object_Trick, $  
          THICK:0, $  
          LINSTYLE:0 }
```

```
END
```

```
.*****  
,
```

Then,

```
IDL> obj = Obj_New('Object_Trick')  
IDL> obj -> GetProperty, Linestyle=linestyle & Print, linestyle  
4  
IDL> obj -> GetProperty, Thick=thick & Print, thick  
2
```

I'll leave it as an exercise for the reader to write the generic SetProperty method. :-)

Cheers,

David

(Hint: Google the IDL archives for SCOPE_VARFETCH.)

--

David Fanning, Ph.D.

Fanning Software Consulting, Inc.

Coyote's Guide to IDL Programming: <http://www.dfanning.com/>

Subject: Re: IDLitComponent - Introspection

Posted by [Robert Barnett](#) on Wed, 11 May 2005 23:48:23 GMT

[View Forum Message](#) <> [Reply to Message](#)

I'm soon to send a feature request to RSI about it. What do you think about the following procedures?

```
;CallGetProperty.pro
;+
; NAME:
;   CallGetProperty
;
; PURPOSE:
;   To get the property of an object, when the keyword is only known at runtime.
;
; CATEGORY:
;   Introspection
;
; CALLING SEQUENCE:
;   CallGetProperty, obj, kwd, value
;
; INPUTS:
;   obj: The object from which to obtain the property
;   kwd: The keyword name of the property
;
; OUTPUTS:
;   value: The value of the property
;
; SIDE EFFECTS:
;   Expects to be able to make a single call "obj -> GetProperty"
;
; EXAMPLE:
;
; ; Define an object
;
; pro boringobj__define
;   struct = {boringobj, boringproperty: ""}
; end
```

```

;
;
; pro boringobj::GetProperty, BORINGPROPERTY=boringproperty
; if (arg_present(boringproperty)) then $
;   boringproperty = self.boringproperty
; end
;
;
; ; Get its property
;
; CallGetProperty, obj_new('boringobj'), 'BORINGPROPERTY', value
; print, value
;
;
;-
pro CallGetProperty, obj, kwd, value
; Unknown implementation
end

;CallSetProperty.pro
;+
; NAME:
;   CallSetProperty
;
; PURPOSE:
;   To set the property of an object, when the keyword is only known at runtime.
;
; CATEGORY:
;   Introspection
;
; CALLING SEQUENCE:
;   CallSetProperty, object, kwd, value
;
; INPUTS:
; object: The object from which to obtain the property
; kwd: The keyword name of the property
; value: The value of the property
;
; SIDE EFFECTS:
; Expects to be able to make a single call "obj -> SetProperty"
;
; EXAMPLE:
;
; ; Define an object
;
; pro boringobj__define
;   struct = {boringobj, boringproperty: ""}
; end
;
;

```

```

; pro boringobj:: SetProperty, BORINGPROPERTY=boringproperty
; if (n_elements(boringproperty) gt 0) then $
;   self.boringproperty = boringproperty
; end
;
; ; Set its property
;
; CallSetProperty, obj_new('boringobj'), 'BORINGPROPERTY', 'boring value'
;
;-
pro CallSetProperty, obj, kwd, value
obj -> SetProperty, _EXTRA=create_struct(kwd, value)
end

```

=====

The actual application I had in mind was a configuration tool.
I have objects with properties which I would like to be able to
edit via a GUI at runtime. I can do this very effectively by using
widget_property sheet.

However, I have numerous objects and do not want to bother setting up a
widget_property sheet for each object individually. Instead, I wish to
query objects for properties which are 'IDLContainer's and
automatically generate a widget_tree based on the links between objects.

--

nrb@ Robbie Barnett
imag Research Assistant
wsahs Nuclear Medicine & Ultrasound
nsw Westmead Hospital
gov Sydney Australia
au +61 2 9845 7223

Subject: Re: IDLitComponent - Introspection
Posted by [marc schellens\[1\]](#) on Thu, 12 May 2005 08:28:55 GMT
[View Forum Message](#) <> [Reply to Message](#)

But what is the problem with execute?
I think this cries for it.
marc

Subject: Re: IDLitComponent - Introspection

Posted by [David Fanning](#) on Thu, 12 May 2005 12:03:21 GMT

[View Forum Message](#) <> [Reply to Message](#)

m_schellens@hotmail.com writes:

- > But what is the problem with execute?
- > I think this cries for it.

Virtual Machine. :-)

Cheers,

David

--

David Fanning, Ph.D.

Fanning Software Consulting, Inc.

Coyote's Guide to IDL Programming: <http://www.dfanning.com/>

Subject: Re: IDLitComponent - Introspection

Posted by [Robert Barnett](#) on Fri, 13 May 2005 05:02:42 GMT

[View Forum Message](#) <> [Reply to Message](#)

Well it is kinda nice that I can run my application on the IDL VM

http://www.zipworld.com.au/~retsil/idl/rt_launcherconfig_test1.sav

This is an example application. It is a configuration tool which allows me to create and maintain a configuration tree according to the relationships between various classes.

It's simply not possible (or at least efficient) to do this kind of thing without introspection. Hence, I've made much use of IDLitComponent::GetPropertyByIdentifier.

Thanks for all your responses.

There are some days when IDL just spoils all the fun and puts up a random brick wall saying "No! You can't do that.". After tearing my hair out over an otherwise insignificant issue I come to this newsgroup for a bit of (syntax) therapy.
I'm very glad that we're all nice to each other round here.

--

Robbie

--
