Subject: Re: iTool identifiers
Posted by Michael Galloy on Wed, 07 Jun 2006 07:47:11 GMT
View Forum Message <> Reply to Message

Robbie wrote:
> Hello,
>
> I was wondering if anyone has used the iTool identifier system just by
> itself. I like the idea of using the iTool identifier system for
> keeping a track of objects, but I am not using iTools for my particular
> project.
>
> By the looks of it, I just have to:
> * Use IDLitContainer instead of IDL_Container
> * Use IDLitComponent as a base class
> * No change necessary for graphics atoms
> and I should be able to reference all objects from the context of the
> root IDL_Container
>
> Is it really that easy? Is it more convenient than using conventional
> object references? Does a global identifier system blur the segregation
> between modules?

I don't think it will be *that* easy.

```
IDL> o = obj_new('idlgrplot', findgen(11))
IDL> print, obj_isa(o, 'idlitcomponent')
   1
IDL> help, o->getFullIdentifier()
<Expression>    STRING    = ''
```

I think you need to start the iTools system and have an iTool for most
components to have a valid identifier (since I think the component gets
its id when it is registered). Maybe there is some way to fake it? To
find out would require digging around the iTools source code.

But, maybe the NAME property for object graphics is good enough?

```
IDL> oview = obj_new('idlgrview')
IDL> omodel = obj_new('idlgrmodel', name='mymodel')
IDL> oview->add, omodel
IDL> oplot = obj_new('idlgrplot', findgen(11), name='myplot')
IDL> omodel->add, oplot
IDL> help, oview->getByName('mymodel/myplot')
<Expression>    OBJREF    = <ObjHeapVar10(IDLGRPLOT)>
```

I use this all the time so that I don't have to store object references
for many different objects in a hierarchy. Is this what you mean by "I

should be able to reference all objects from the context of the root
IDL_Container"?

Mike
--
www.michaelgalloy.com

Subject: Re: iTool identifiers
Posted by Chris[2] on Wed, 07 Jun 2006 23:46:09 GMT

View Forum Message <> Reply to Message

Actually, I think it *is* that easy:

```
IDL> o=obj_new('idlitcomponent', identifier='myobject')
IDL> oparent = obj_new('idlitcontainer', identifier='parent')

IDL> ograndparent = obj_new('idlitcontainer')

IDL> oparent->add, o

IDL> ograndparent->add, oparent

IDL> print, o->getfullidentifier()

/PARENT/MYOBJECT

IDL> help, ograndparent->getbyidentifier('/parent/myobject')

<Expression> OBJREF = <ObjHeapVar10(IDLITCOMPONENT)>
```

The key is to set the IDENTIFIER keyword when you create your objects. As long as your objects are a subclass of IDLitComponent, and your containers are IDLitContainer, then everything should just work, and you shouldn't need any of the iTools framework.

-Chris

ITT Visual Information Solutions

"Michael Galloy" <mgalloy@gmail.com> wrote in message
news:3wvhg.10064$921.7004@newsread4.news.pas.earthlink.net.. .
> Robbie wrote:
>> Hello,
>>
>> I was wondering if anyone has used the iTool identifier system just by

>> itself. I like the idea of using the iTool identifier system for
>> keeping a track of objects, but I am not using iTools for my particular
>> project.
>>
>> By the looks of it, I just have to:
>> * Use IDLitContainer instead of IDL_Container
>> * Use IDLitComponent as a base class
>> * No change necessary for graphics atoms
>> and I should be able to reference all objects from the context of the
>> root IDL_Container
>>
>> Is it really that easy? Is it more convenient than using conventional
>> object references? Does a global identifier system blur the segregation
>> between modules?
>
> I don't think it will be *that* easy.
>
> IDL> o = obj_new('idlgrplot', findgen(11))
> IDL> print, obj_isa(o, 'idlitcomponent')
>   1
> IDL> help, o->getFullIdentifier()
> <Expression>   STRING   = ''
>
> I think you need to start the iTools system and have an iTool for most
> components to have a valid identifier (since I think the component gets
> its id when it is registered). Maybe there is some way to fake it? To find
> out would require digging around the iTools source code.
>
> But, maybe the NAME property for object graphics is good enough?
>
> IDL> oview = obj_new('idlgrview')
> IDL> omodel = obj_new('idlgrmodel', name='mymodel')
> IDL> oview->add, omodel
> IDL> oplot = obj_new('idlgrplot', findgen(11), name='myplot')
> IDL> omodel->add, oplot
> IDL> help, oview->getByName('mymodel/myplot')
> <Expression>   OBJREF   = <ObjHeapVar10(IDLGRPLOT)>
>
> I use this all the time so that I don't have to store object references
> for many different objects in a hierarchy. Is this what you mean by "I
> should be able to reference all objects from the context of the root
> IDL_Container"?
>
> Mike
> --
> www.michaelgalloy.com

Subject: Re: iTool identifiers
Posted by Michael Galloy on Thu, 08 Jun 2006 03:58:27 GMT
View Forum Message <> Reply to Message

Chris Torrence wrote:

```
>  Actually, I think it *is* that easy:
>
>  IDL> o=obj_new('idlitcomponent', identifier='myobject')
>  IDL> oparent = obj_new('idlitcontainer', identifier='parent')
>
>  IDL> ograndparent = obj_new('idlitcontainer')
>
>  IDL> oparent->add, o
>
>  IDL> ograndparent->add, oparent
>
>  IDL> print, o->getfullidentifier()
>
>  /PARENT/MYOBJECT
>
>  IDL> help, ograndparent->getbyidentifier('/parent/myobject')
>
>  <Expression> OBJREF = <ObjHeapVar10(IDLITCOMPONENT)>
>
>  The key is to set the IDENTIFIER keyword when you create your objects. As
>  long as your objects are a subclass of IDLitComponent, and your containers
>  are IDLitContainer, then everything should just work, and you shouldn't need
>  any of the iTools framework.
>
>  -Chris
>
>  ITT Visual Information Solutions
```

Cool. And IDLitContainer::findIdentifiers has a nice searching capability.

Mike
--
www.michaelgalloy.com

Subject: Re: iTool identifiers
Posted by Robbie on Wed, 14 Jun 2006 06:13:29 GMT
View Forum Message <> Reply to Message

It's a pity IDLitContainer::findIdentifiers doesn't search through the
object graphics tree. I've been trying to find neat ways of creating
and maintaining an object graphics tree.

I'm in the process of writing a class to create graphics objects using methods which are a bit like tv and plot.

http://www.barnett.id.au/idl/