Subject: Object Field Name Conflicts
Posted by David Fanning on Sun, 26 Jan 2014 02:52:32 GMT
View Forum Message <> Reply to Message

Folks,

In building an object program recently I have a core object that
inherits IDL_CONTAINER, IDLitCOMPONENT, and IDL_OBJECT objects. I don't
know the field or tag names of *any* of these objects, which makes
choosing names for my own object a bit of an adventure.

I've already run into conflict with _parent, _name, and _uvalue. Of
course, this is not surprising. Great minds think alike and all that.
But, I wonder if there is a listing of the tag names these objects use?

It would be wonderful if the name space of objects could be protected,
but failing that, it would be great if we knew what names we can't use.

Cheers,

David
--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
Sepore ma de ni thue. ("Perhaps thou speakest truth.")

Subject: Re: Object Field Name Conflicts
Posted by David Fanning on Sun, 26 Jan 2014 03:06:19 GMT
View Forum Message <> Reply to Message

David Fanning writes:

> In building an object program recently I have a core object that
> inherits IDL_CONTAINER, IDLitCOMPONENT, and IDL_OBJECT objects. I don't
> know the field or tag names of *any* of these objects, which makes
> choosing names for my own object a bit of an adventure.
>
> I've already run into conflict with _parent, _name, and _uvalue. Of
> course, this is not surprising. Great minds think alike and all that.
> But, I wonder if there is a listing of the tag names these objects use?
>
> It would be wonderful if the name space of objects could be protected,
> but failing that, it would be great if we knew what names we can't use.

You may be thinking, "Who cares!? I don't see why this is important."

Here is an example. Presumably the uvalue of one of these objects is whatever I want it to be. It certainly works that way. So, I guess it must be something that looks like a pointer. But, I want to have a user value that I can transfer data into without making an extra copy of the data. In other words, if I were going to put this in a pointer, I would do this:

```
  uvalue = Ptr_New(bigData, /No_Copy)
```

Is this possible with the uvalue in IDLitComponent (where I suspect this is located)? If so, how would I do it?

Cheers,

David
--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
Sepore ma de ni thue. ("Perhaps thou speakest truth.")

Subject: Re: Object Field Name Conflicts
Posted by Michael Galloy on Sun, 26 Jan 2014 03:29:01 GMT

View Forum Message <> Reply to Message

On 1/25/14, 7:52 PM, David Fanning wrote:
> Folks,
>
> In building an object program recently I have a core object that
> inherits IDL_CONTAINER, IDLitCOMPONENT, and IDL_OBJECT objects. I don't
> know the field or tag names of *any* of these objects, which makes
> choosing names for my own object a bit of an adventure.
>
> I've already run into conflict with _parent, _name, and _uvalue. Of
> course, this is not surprising. Great minds think alike and all that.
> But, I wonder if there is a listing of the tag names these objects use?
>
> It would be wonderful if the name space of objects could be protected,
> but failing that, it would be great if we knew what names we can't use.
>
> Cheers,
>
> David
>

```
IDL> help, { idl_object }
** Structure IDL_OBJECT, 3 tags, length=24, data length=20:
   IDL_OBJECT_TOP  LONG64                     0
   __OBJ__         OBJREF    <NullObject>
   IDL_OBJECT_BOTTOM
                   LONG64                     0
IDL> help, { idl_container }
** Structure IDL_CONTAINER, 6 tags, length=32, data length=30:
   IDL_CONTAINER_TOP
                   LONG64                     0
   IDLCONTAINERVERSION
                   INT              0
   PHEAD           POINTER   <NullPointer>
   PTAIL           POINTER   <NullPointer>
   NLIST           LONG                 0
   IDL_CONTAINER_BOTTOM
                   LONG64                     0
IDL> help, { idlitcomponent }
** Structure IDLITCOMPONENT, 14 tags, length=128, data length=122:
   IDLITCOMPONENT_TOP
                   LONG64                     0
   IDLITCOMPONENTVERSION
                   INT              0
   DESCRIPTION     STRING    ''
   NAME            STRING    ''
   ICON            STRING    ''
   IDENTIFIER      STRING    ''
   HELP            STRING    ''
   TOOL            OBJREF    <NullObject>
   UVALUE          POINTER   <NullPointer>
   _PARENT         OBJREF    <NullObject>
   _PROXY          OBJREF    <NullObject>
   PROPERTYDESCRIPTORS
                   OBJREF    <NullObject>
   _FLAGS          LONG                 0
   IDLITCOMPONENT_BOTTOM
                   LONG64                     0
```

Mike
--
Michael Galloy
www.michaelgalloy.com
Modern IDL: A Guide to IDL Programming (http://modernidl.idldev.com)
Research Mathematician
Tech-X Corporation

Subject: Re: Object Field Name Conflicts
Posted by David Fanning on Sun, 26 Jan 2014 05:28:42 GMT

View Forum Message <> Reply to Message

Michael Galloy writes:

```
>  IDL> help, { idl_object }
>  ** Structure IDL_OBJECT, 3 tags, length=24, data length=20:
>     IDL_OBJECT_TOP  LONG64                      0
>     __OBJ__         OBJREF    <NullObject>
>     IDL_OBJECT_BOTTOM
>                 LONG64                      0
```

Woohoo! Thanks.

Is this another example of operator overloading?

Cheers,

David
--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
Sepore ma de ni thue. ("Perhaps thou speakest truth.")

Subject: Re: Object Field Name Conflicts
Posted by Michael Galloy on Sun, 26 Jan 2014 15:33:18 GMT

View Forum Message <> Reply to Message

David Fanning <news@idlcoyote.com> wrote:
> Michael Galloy writes:
>

```
>>>  help, { idl_object }
>>  ** Structure IDL_OBJECT, 3 tags, length=24, data length=20:
>>     IDL_OBJECT_TOP  LONG64                      0
>>     __OBJ__         OBJREF    <NullObject>
>>     IDL_OBJECT_BOTTOM
>>                 LONG64                      0
```

>
> Woohoo! Thanks.
>
> Is this another example of operator overloading?
>
> Cheers,
>
> David

I don't believe so. IDL objects have always been a named structure definition plus more.

Mike
--
www.michaelgalloy.com
Research Mathematician
Tech-X Corporation

Subject: Re: Object Field Name Conflicts
Posted by David Fanning on Sun, 26 Jan 2014 15:41:57 GMT
View Forum Message <> Reply to Message

Michael Galloy writes:

```
>
>  David Fanning <news@idlcoyote.com> wrote:
>>  Michael Galloy writes:
>>
>>>>  help, { idl_object }
>>>  ** Structure IDL_OBJECT, 3 tags, length=24, data length=20:
>>>      IDL_OBJECT_TOP  LONG64                         0
>>>      __OBJ__         OBJREF    <NullObject>
>>>      IDL_OBJECT_BOTTOM
>>>                      LONG64                         0
>>
>>  Woohoo! Thanks.
>>
>>  Is this another example of operator overloading?
>>
>>  Cheers,
>>
>>  David
>
>  I don't believe so. IDL objects have always been a named structure
>  definition plus more.
```

I wonder if someone can check this for us in IDL 7, just to be sure.

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
Sepore ma de ni thue. ("Perhaps thou speakest truth.")

Subject: Re: Object Field Name Conflicts
Posted by Lajos Foldy on Sun, 26 Jan 2014 16:07:19 GMT
View Forum Message <> Reply to Message

On Sunday, January 26, 2014 4:41:57 PM UTC+1, David Fanning wrote:

```
>>>> > help, { idl_object }
>>>>  ** Structure IDL_OBJECT, 3 tags, length=24, data length=20:
>>>>     IDL_OBJECT_TOP  LONG64                          0
>>>>     __OBJ__         OBJREF    <NullObject>
>>>>     IDL_OBJECT_BOTTOM
>>>>                     LONG64                          0
>
>  I wonder if someone can check this for us in IDL 7, just to be sure.
>
```

```
IDL> print, !version
{ sparc sunos unix 5.3 Nov 11 1999}
IDL> help, {idl_container}, /str
** Structure IDL_CONTAINER, 6 tags, length=32:
   IDL_CONTAINER_TOP
                   DOUBLE    =       0.0000000
   IDLCONTAINERVERSION
                   INT       =        0
   PHEAD           POINTER   <NullPointer>
   PTAIL           POINTER   <NullPointer>
   NLIST           LONG                 0
   IDL_CONTAINER_BOTTOM
                   DOUBLE    =       0.0000000
IDL>
```

(I think IDL_OBJECT was introduced in IDL8).

regards,
Lajos

Subject: Re: Object Field Name Conflicts
Posted by David Fanning on Sun, 26 Jan 2014 16:09:29 GMT
View Forum Message <> Reply to Message

fawltylanguage@gmail.com writes:

```
>
>  On Sunday, January 26, 2014 4:41:57 PM UTC+1, David Fanning wrote:
>
>>>> >> help, { idl_object }
>>>> > ** Structure IDL_OBJECT, 3 tags, length=24, data length=20:
```

```
>>>> >    IDL_OBJECT_TOP  LONG64                      0
>>>> >    __OBJ__         OBJREF    <NullObject>
>>>> >    IDL_OBJECT_BOTTOM
>>>> >                    LONG64                      0
>>
>>  I wonder if someone can check this for us in IDL 7, just to be sure.
>>
>
>  IDL> print, !version
>  { sparc sunos unix 5.3 Nov 11 1999}
>  IDL> help, {idl_container}, /str
>  ** Structure IDL_CONTAINER, 6 tags, length=32:
>     IDL_CONTAINER_TOP
>                     DOUBLE    =       0.0000000
>     IDLCONTAINERVERSION
>                     INT       =       0
>     PHEAD           POINTER   <NullPointer>
>     PTAIL           POINTER   <NullPointer>
>     NLIST           LONG                 0
>     IDL_CONTAINER_BOTTOM
>                     DOUBLE    =       0.0000000
>  IDL>
>
>  (I think IDL_OBJECT was introduced in IDL8).
```

Yikes! What you learn about IDL every day! ;-)

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
Sepore ma de ni thue. ("Perhaps thou speakest truth.")

Subject: Re: Object Field Name Conflicts
Posted by Phillip Bitzer on Sun, 26 Jan 2014 21:28:26 GMT

View Forum Message <> Reply to Message

On Sunday, January 26, 2014 10:07:19 AM UTC-6, fawltyl...@gmail.com wrote:
>
>
> (I think IDL_OBJECT was introduced in IDL8).
>

Yep: http://www.exelisvis.com/docs/IDL_Object.html

On Sunday, January 26, 2014 10:09:29 AM UTC-6, David Fanning wrote:

>
> Yikes! What you learn about IDL every day! ;-)
>

No kidding. It doesn't surprise me, but I definitely didn't know about the help, {obj_name} to get the "tags" of an object.