
Subject: Re: Is there a way to use C++ code in IDL
Posted by [David Fanning](#) on Thu, 20 Feb 2003 16:48:33 GMT
[View Forum Message](#) <> [Reply to Message](#)

CelticBlues (idluser@celticblues.com) writes:

> I have an ascii data file that I need to read. I already have C++ code that
> will parse the file and store it in a structure. Is there a way to reuse
> this C++ code in IDL so that I don't have to recode the parse routines?

Yes, there is a way to reuse this code, but learning the methods to do it will far surpass (by many orders of magnitude) the time it will take (say 5 minutes) to learn to parse an ASCII file in IDL. :-)

See Call_External, Make_DLL, LinkImage, etc.

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: Is there a way to use C++ code in IDL
Posted by [James Kuyper](#) on Thu, 20 Feb 2003 17:08:27 GMT
[View Forum Message](#) <> [Reply to Message](#)

David Fanning wrote:

>
> CelticBlues (idluser@celticblues.com) writes:
>
>> I have an ascii data file that I need to read. I already have C++ code that
>> will parse the file and store it in a structure. Is there a way to reuse
>> this C++ code in IDL so that I don't have to recode the parse routines?
>
> Yes, there is a way to reuse this code, but learning
> the methods to do it will far surpass (by many orders of
> magnitude) the time it will take (say 5 minutes) to
> learn to parse an ASCII file in IDL. :-)

That depends entirely on how complicated the syntax that it parses is. If he's worried about the recoding effort, I doubt that the parsing is simple enough to be a 5 minute job in IDL. With all due respect to IDL,

most things that can be parsed correctly by a 5-minute IDL program have a fairly simple syntax.

Subject: Re: Is there a way to use C++ code in IDL
Posted by [David Fanning](#) on Thu, 20 Feb 2003 17:54:10 GMT
[View Forum Message](#) <> [Reply to Message](#)

James Kuyper (kuyper@saicmodis.com) writes:

> That depends entirely on how complicated the syntax that it parses is.
> If he's worried about the recoding effort, I doubt that the parsing is
> simple enough to be a 5 minute job in IDL. With all due respect to IDL,
> most things that can be parsed correctly by a 5-minute IDL program have
> a fairly simple syntax.

Perhaps I exaggerated a little. (It's been known to happen.)
But I stand by my "orders of magnitude" comment. :-)

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: Is there a way to use C++ code in IDL
Posted by [CelticBlues](#) on Thu, 20 Feb 2003 18:12:34 GMT
[View Forum Message](#) <> [Reply to Message](#)

It's a fairly complex ascii file, the parser of which I did not write myself. Combine the complexity, with the fact that I didn't write the original parser, and I feel more comfortable reusing the code, than rewriting it in IDL, which I just started using yesterday. Maybe as I learn more about IDL, I'll write a parser in IDL, but until then....

Thanks for the info,

Later,
Ed

"David Fanning" <david@dfanning.com> wrote in message
news:MPG.18bec429c19cf041989af0@news.frii.com...

> James Kuyper (kuyper@saicmodis.com) writes:

>

>> That depends entirely on how complicated the syntax that it parses is.

>> If he's worried about the recoding effort, I doubt that the parsing is

>> simple enough to be a 5 minute job in IDL. With all due respect to IDL,

>> most things that can be parsed correctly by a 5-minute IDL program have

>> a fairly simple syntax.

>

> Perhaps I exaggerated a little. (It's been known to happen.)

> But I stand by my "orders of magnitude" comment. :-)

>

> 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: Is there a way to use C++ code in IDL

Posted by [Paul Van Delst\[1\]](#) on Thu, 20 Feb 2003 18:45:42 GMT

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

CelticBlues wrote:

>

> It's a fairly complex ascii file, the parser of which I did not write

> myself. Combine the complexity, with the fact that I didn't write the

> original parser, and I feel more comfortable reusing the code, than

> rewriting it in IDL, which I just started using yesterday. Maybe as I learn

> more about IDL, I'll write a parser in IDL, but until then....

I'm not being flippant when I say this may be a good opportunity to learn IDL - and understand more about your data file/structure. Assuming you have the luxury of time.

Otherwise, why not use the C++ parser to read the file and write some simple little C++ program to output the structure in some really simple format that would take "5 minutes" to write a parser for to read into IDL? (Here I'm assuming all you want to use IDL for in this case is plot the data)

paulv

>
> Thanks for the info,
>
> Later,
> Ed
>
> "David Fanning" <david@dfanning.com> wrote in message
> news:MPG.18bec429c19cf041989af0@news.frii.com...
>> James Kuyper (kuyper@saicmodis.com) writes:
>>
>>> That depends entirely on how complicated the syntax that it parses is.
>>> If he's worried about the recoding effort, I doubt that the parsing is
>>> simple enough to be a 5 minute job in IDL. With all due respect to IDL,
>>> most things that can be parsed correctly by a 5-minute IDL program have
>>> a fairly simple syntax.
>>
>> Perhaps I exaggerated a little. (It's been known to happen.)
>> But I stand by my "orders of magnitude" comment. :-)
>>
>> 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

--
Paul van Delst
CIMSS @ NOAA/NCEP/EMC
Ph: (301)763-8000 x7274
Fax:(301)763-8545

Subject: Re: Is there a way to use C++ code in IDL
Posted by [CelticBlues](#) on Thu, 20 Feb 2003 18:51:27 GMT
[View Forum Message](#) <> [Reply to Message](#)

The file is a home grown tagged file format containing a type of image data and additional information about the conditions at the time the image was taken etc. Here is an example of what a simple file may look like:

[System]
 Name = 1
[End]

[Data Sets] // will be at least one data set, could be as many as 3

 DataSet1

[End]

[Files Attributes]

 [Conditions File]

 [Properties]

 Name = 97918.10

 CreateTime = Thu Sept 18 1997 at 11:21:36

 LastAccessTime = Mon Jan 01 1601

 LastModifiedTime = Thu Sept 18 1997 at 11:21:36

 Size = 16882000 // bytes

 [End]

 [Frame information]

 Width = 133

 Height = 148

 [End]

 [End] // end [Telemetry File]

[Saved File]

 CreateDate = Tue Feb 11 2003 at 16:08:54

 Width = 133

 Height = 148

 FrameNumber = 1

 DataUnits = meters

 Comment =

 CoordinateSystem = UTM // Latlong or UTM

[End]

[End] // end [Files Attributes]

[DataBlock:DataSet1]

 R1P1 R1P2...R1P148

 R2P1 R2P2...R2P148

 .

 .

 .

 R133P1 R133P2...R133P148

[End]

R1P1 = Row 1 Pixel 1, R2P1 = Row 2, Pixel 2 etc.

One additional bit of info... the datablock may not actually have CR/NL
after each row, i.e. it may be

[DataBlock:DataSet1]

 R1P1 R1P2...R1P148 R2P1 R2P2...R2P148 ... R133P1 R133P2...R133P148

[End]

Ok, assume that I want to rewrite the parser in IDL... From what I understand it is simple enough to read an ascii table from a file, but what about reading this type of format?

Ed

"David Fanning" <david@dfanning.com> wrote in message
news:MPG.18beb4cde3551d28989aef@news.frii.com...

> CelticBlues (idluser@celticblues.com) writes:

>

>> I have an ascii data file that I need to read. I already have C++ code
that

>> will parse the file and store it in a structure. Is there a way to
reuse

>> this C++ code in IDL so that I don't have to recode the parse routines?

>

> Yes, there is a way to reuse this code, but learning
> the methods to do it will far surpass (by many orders of
> magnitude) the time it will take (say 5 minutes) to
> learn to parse an ASCII file in IDL. :-)

>

> See Call_External, Make_DLL, LinkImage, etc.

>

> 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: Is there a way to use C++ code in IDL
Posted by [David Fanning](#) on Thu, 20 Feb 2003 18:52:14 GMT

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

CelticBlues (idluser@celticblues.com) writes:

> It's a fairly complex ascii file, the parser of which I did not write
> myself. Combine the complexity, with the fact that I didn't write the
> original parser, and I feel more comfortable reusing the code, than
> rewriting it in IDL, which I just started using yesterday. Maybe as I learn
> more about IDL, I'll write a parser in IDL, but until then....

My only point is that learning how to link C++ programs into IDL on your second day on the job, is like learning C++ by starting with how the compiler works. It may be useful (and even interesting), but at the end of the day you aren't going to be much of anywhere.

But if you spend a day with IDL futzing around trying to read an ASCII file, you might even have a file reader by the time the dinner bell goes off. :-)

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: Is there a way to use C++ code in IDL
Posted by [CelticBlues](#) on Thu, 20 Feb 2003 19:11:39 GMT
[View Forum Message](#) <> [Reply to Message](#)

For me, it would be ideal, if I could do everything in IDL...but I don't work in an ideal world... So I think your suggestion of an intermediate file written by the original parser is probably the way to go, although I still have the IDL parser on the back burner. That will give me a fall back, if I can't get an ascii reader written in IDL by dinner time (as David Fanning suggests :-)...).

Thanks to everyone for the comments,
Ed

"Paul van Delst" <paul.vandelst@noaa.gov> wrote in message
news:3E552256.3BF2399F@noaa.gov...

> CelticBlues wrote:

>>

>> It's a fairly complex ascii file, the parser of which I did not write
>> myself. Combine the complexity, with the fact that I didn't write the
>> original parser, and I feel more comfortable reusing the code, than
>> rewriting it in IDL, which I just started using yesterday. Maybe as I
learn

>> more about IDL, I'll write a parser in IDL, but until then....

>

> I'm not being flippant when I say this may be a good opportunity to learn IDL - and
> understand more about your data file/structure. Assuming you have the luxury of time.
>
> Otherwise, why not use the C++ parser to read the file and write some simple little C++
> program to output the structure in some really simple format that would take "5 minutes"
> to write a parser for to read into IDL? (Here I'm assuming all you want to use IDL for in
> this case is plot the data)
>
> paulv
>
>
>>
>> Thanks for the info,
>>
>> Later,
>> Ed
>>
>> "David Fanning" <david@dfanning.com> wrote in message
>> news:MPG.18bec429c19cf041989af0@news.frii.com...
>>> James Kuyper (kuyper@saicmodis.com) writes:
>>>
>>>> That depends entirely on how complicated the syntax that it parses is.
>>>> If he's worried about the recoding effort, I doubt that the parsing is
>>>> simple enough to be a 5 minute job in IDL. With all due respect to IDL,
>>>> most things that can be parsed correctly by a 5-minute IDL program have
>>>> a fairly simple syntax.
>>>
>>> Perhaps I exaggerated a little. (It's been known to happen.)
>>> But I stand by my "orders of magnitude" comment. :-)
>>>
>>> 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
>
> --
> Paul van Delst
> CIMSS @ NOAA/NCEP/EMC
> Ph: (301)763-8000 x7274
> Fax:(301)763-8545

Subject: Re: Is there a way to use C++ code in IDL
Posted by [David Fanning](#) on Thu, 20 Feb 2003 19:16:43 GMT
[View Forum Message](#) <> [Reply to Message](#)

CelticBlues (idluser@celticblues.com) writes:

> The file is a home grown tagged file format containing a type of image data
> and additional information about the conditions at the time the image was
> taken etc. Here is an example of what a simple file may look like:

Oh, no. I'm not going to be sucked into this. I have my own work to do. (Actually, now that I think about it, I am a little short of paying customers at the moment...) But, no, I'm not going to do this.

This is not a hard file format to figure out.
Learn how to read a line of the file at a time:

```
line = ""  
OPENR, lun, 'myfile.dat', /Get_Lun  
ReadF, lun, line
```

Learn how to parse that line. I'd pay particular attention to STRPOS and STRMID, for starters. I'd learn what FIX, LONG, and FLOAT can do.

I don't know what kind of data that is in the data block, but presumably those letters mean *something*. I'd learn how to decode that.

There is a day's worth of material here, for sure. But this is a LOT easier than you are probably thinking right now. Try it out by doing things at the IDL command line. IDL is fabulous at what I call "learning by noodling around". You have a tremendous advantage of being able to see and make sense of what is in the data file. We are not all so lucky.

And, for heaven's sake, get a book about IDL programming.
If you are a serious sort of person get Liam Gumley's book. If you prefer something with a bit more attitude, get my book. But you are unlikely to learn what you really want to know by looking at the IDL documentation. Either one of these books will parse the essential from the overwhelming for you. :-)

Good luck. I think you are going to find out you LOVE IDL (well, eventually). :-)

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: Is there a way to use C++ code in IDL
Posted by [Rick Towler](#) on Thu, 20 Feb 2003 19:17:16 GMT
[View Forum Message](#) <> [Reply to Message](#)

"CelticBlues" wrote in

> It's a fairly complex ascii file, the parser of which I did not write
> myself. Combine the complexity, with the fact that I didn't write the
> original parser, and I feel more comfortable reusing the code, than
> rewriting it in IDL, which I just started using yesterday. Maybe as I
learn
> more about IDL, I'll write a parser in IDL, but until then....

Everyone seems to be chiming in with their opinions and they are all valid.

Determining where to spend your effort is difficult, especially if you don't have much experience with IDL and you don't have intimate knowledge of the file format. You can really help your cause if you post a snippet of your file since some of the experienced IDL'ers can give you their opinion on the level of effort needed to parse a record in IDL.

If you decide to go the C++ route I would suggest coding a wrapper function in C++ to interface your code to IDL as an dynamically loadable module (aka dlm). I know you don't have much experience with IDL. I am hoping you have

some with C/C++? As a novice C/C++ coder I have written a number of simple .dlms. If you have experience with C/C++ you should be well on your way.

How simple is it to use the C++ code? Do you just pass the function a file name and it returns a struct or do you pass it a record as a char array and it returns the parsed record in a struct? The simpler to use, the simpler to code into a dlm.

Also, if you go the dlm route I strongly suggest picking up a copy of Ronn Kling's book "Calling C from IDL" available from his website www.kilvarock.com.

I could probably work up an example of a dlm which returns a structure to IDL if you are interested.

-Rick

Subject: Re: Is there a way to use C++ code in IDL
Posted by [Paul Van Delst\[1\]](#) on Thu, 20 Feb 2003 19:28:06 GMT
[View Forum Message](#) <> [Reply to Message](#)

CelticBlues wrote:

```
>
> The file is a home grown tagged file format containing a type of image data
> and additional information about the conditions at the time the image was
> taken etc. Here is an example of what a simple file may look like:
> -----
> [System]
>   Name = 1
> [End]
> [Data Sets] // will be at least one data set, could be as many as 3
>   DataSet1
> [End]
> [Files Attributes]
>   [Conditions File]
>     [Properties]
>       Name = 97918.10
>       CreateTime = Thu Sept 18 1997 at 11:21:36
>       LastAccessTime = Mon Jan 01 1601
>       LastModifiedTime = Thu Sept 18 1997 at 11:21:36
>       Size = 16882000    // bytes
>     [End]
>   [Frame information]
>     Width = 133
>     Height = 148
>   [End]
> [End]    // end [Telemetry File]
```

```

>
> [Saved File]
>   CreateDate = Tue Feb 11 2003 at 16:08:54
>   Width = 133
>   Height = 148
>   FrameNumber = 1
>   DataUnits = meters
>   Comment =
>   CoordinateSystem = UTM    // Latlong or UTM
> [End]
>
> [End]    // end [Files Attributes]
>
> [DataBlock:DataSet1]
>   R1P1 R1P2...R1P148
>   R2P1 R2P2...R2P148
>   .
>   .
>   .
>   R133P1 R133P2...R133P148
> [End]
>
> -----
>
> R1P1 = Row 1 Pixel 1, R2P1 = Row 2, Pixel 2 etc.
>
> One additional bit of info... the datablock may not actually have CR/NL
> after each row, i.e. it may be
>
> [DataBlock:DataSet1]
>   R1P1 R1P2...R1P148 R2P1 R2P2...R2P148 ... R133P1 R133P2...R133P148
> [End]
>
> Ok, assume that I want to rewrite the parser in IDL... From what I
> understand it is simple enough to read an ascii table from a file, but what
> about reading this type of format?

```

The reading isn't the problem - it's the sorting of the read-in bits. :o)

But, it doesn't look too difficult. Given what you know about the allowable syntax in your ASCII file, you can create the structures up front, e.g.

```

Properties = { Name:      0.0, $
               CreateTime:  '', $
               LastAccessTime:  '', $
               LastModifiedTime:  '', $
               Size:      0L }

```

```
FrameInformation = { Width: 0L, $  
                    Height: 0L }
```

```
ConditionsFile = { Properties: Properties, $  
                  Frame_Information: FrameInformation }
```

```
SavedFile = { ...etc }
```

```
FilesAttributes = { ConditionsFile: ConditionsFile, $  
                   SavedFile: SavedFile }
```

etc..etc.

although your lines may get a bit long, e.g.

```
print, FilesAttributes.ConditionsFiles.Properties.LastModifiedTime
```

...phew!

I would use structure constructor functions (search for "Automatic Structure Definition" in the IDL help) to do it, so you can automatically define a valid structure containing all the allowed bits. You could also approach this in an OO way (search for "IDL Object Overview"). I'm not an IDL OO feller, so others will have more to say about that aspect.

All the data stuff would be stored in a pointer array structure element that you can allocate when you determine what the required dimensions are.

Actually it sounds like a neat little project to do this. you'll learn everything you ever wanted to know about IDL structures, and if you're keen, about the OO capabilities also.

paulv

--
Paul van Delst
CIMSS @ NOAA/NCEP/EMC
Ph: (301)763-8000 x7274
Fax:(301)763-8545

Subject: Re: Is there a way to use C++ code in IDL
Posted by [CelticBlues](#) on Thu, 20 Feb 2003 19:54:18 GMT
[View Forum Message](#) <> [Reply to Message](#)

See below....

"David Fanning" <david@dfanning.com> wrote in message

news:MPG.18bed78a6fdf4fbf989af2@news.frii.com...

> CelticBlues (idluser@celticblues.com) writes:

>

>> The file is a home grown tagged file format containing a type of image data

>> and additional information about the conditions at the time the image was

>> taken etc. Here is an example of what a simple file may look like:

>

> Oh, no. I'm not going to be sucked into this. I have my
> own work to do. (Actually, now that I think about it, I
> am a little short of paying customers at the moment...)
> But, no, I'm not going to do this.

Darn... you mean you won't write it for me? ;-) Actually the info you give below is just what I need... An idea of where to look for the answer. I don't expect anyone to hand me the complete solution (where's the fun in that) but a little push in the right direction...

Thanks,

Ed

> And, for heaven's sake, get a book about IDL programming.
> If you are a serious sort of person get Liam Gumley's
> book. If you prefer something with a bit more attitude,
> get my book. But you are unlikely to learn what you
> really want to know by looking at the IDL documentation.
> Either one of these books will parse the essential from
> the overwhelming for you. :-)

Not sure what you mean by "a serious sort of person"... but I like books that are somewhat tutorial like, but aren't limited to overly simple examples. The OpenGL Red Book a good example of the type of book I like to learn a new topic with....

Later,

Ed

Subject: Re: Is there a way to use C++ code in IDL
Posted by [CelticBlues](#) on Thu, 20 Feb 2003 19:55:34 GMT
[View Forum Message](#) <> [Reply to Message](#)

Good stuff... thanks for the info on structs...
Ed

"Paul van Delst" <paul.vandelst@noaa.gov> wrote in message
news:3E552C46.ECC98EAB@noaa.gov...
> CelticBlues wrote:

```

>>
>> The file is a home grown tagged file format containing a type of image
data
>> and additional information about the conditions at the time the image
was
>> taken etc. Here is an example of what a simple file may look like:
>> -----
>> [System]
>>   Name = 1
>> [End]
>> [Data Sets] // will be at least one data set, could be as many as 3
>>   DataSet1
>> [End]
>> [Files Attributes]
>>   [Conditions File]
>>     [Properties]
>>       Name = 97918.10
>>       CreateTime = Thu Sept 18 1997 at 11:21:36
>>       LastAccessTime = Mon Jan 01 1601
>>       LastModifiedTime = Thu Sept 18 1997 at 11:21:36
>>       Size = 16882000    // bytes
>>     [End]
>>     [Frame information]
>>       Width = 133
>>       Height = 148
>>     [End]
>>   [End]    // end [Telemetry File]
>>
>> [Saved File]
>>   CreateDate = Tue Feb 11 2003 at 16:08:54
>>   Width = 133
>>   Height = 148
>>   FrameNumber = 1
>>   DataUnits = meters
>>   Comment =
>>   CoordinateSystem = UTM    // Latlong or UTM
>> [End]
>>
>> [End]    // end [Files Attributes]
>>
>> [DataBlock:DataSet1]
>>   R1P1 R1P2...R1P148
>>   R2P1 R2P2...R2P148
>>   .
>>   .
>>   .
>>   R133P1 R133P2...R133P148
>> [End]

```

```

>>
>> -----
>>
>> R1P1 = Row 1 Pixel 1, R2P1 = Row 2, Pixel 2 etc.
>>
>> One additional bit of info... the datablock may not actually have CR/NL
>> after each row, i.e. it may be
>>
>> [DataBlock:DataSet1]
>>   R1P1 R1P2...R1P148 R2P1 R2P2...R2P148 ... R133P1 R133P2...R133P148
>> [End]
>>
>> Ok, assume that I want to rewrite the parser in IDL... From what I
>> understand it is simple enough to read an ascii table from a file, but
>> what
>> about reading this type of format?
>
> The reading isn't the problem - it's the sorting of the read-in bits. :o)
>
> But, it doesn't look too difficult. Given what you know about the
> allowable syntax in your
> ASCII file, you can create the structures up front, e.g.
>
> Properties = { Name:      0.0, $
>               CreateTime:  '', $
>               LastAccessTime:  '', $
>               LastModifiedTime:  '', $
>               Size:      0L }
>
> FrameInformation = { Width: 0L, $
>                    Height: 0L }
>
> ConditionsFile = { Properties: Properties, $
>                  Frame_Information: FrameInformation }
>
> SavedFile = { ...etc }
>
>
> FilesAttributes = { ConditionsFile: ConditionsFile, $
>                   SavedFile: SavedFile }
>
> etc..etc.
>
> although your lines may get a bit long, e.g.
>
> print, FilesAttributes.ConditionsFiles.Properties.LastModifiedTime
>
> ...pew!

```


>
> I would use structure constructor functions (search for "Automatic
Structure Definition" in
> the IDL help) to do it, so you can automatically define a valid structure
containing all
> the allowed bits. You could also approach this in an OO way (search for
"IDL Object
> Overview"). I'm not an IDL OO feller, so others will have more to say
about that aspect.
>
> All the data stuff would be stored in a pointer array structure element
that you can
> allocate when you determine what the required dimensions are.
>
> Actually it sounds like a neat little project to do this. you'll learn
everything you
> ever wanted to know about IDL structures, and if you're keen, about the OO
capabilities
> also.
>
> paulv
>
> --
> Paul van Delst
> CIMSS @ NOAA/NCEP/EMC
> Ph: (301)763-8000 x7274
> Fax:(301)763-8545

Subject: Re: Is there a way to use C++ code in IDL
Posted by [Rick Towler](#) on Thu, 20 Feb 2003 20:07:38 GMT
[View Forum Message](#) <> [Reply to Message](#)

Here is a pile of bad code for you :)

I wrote a file parser for some home grown tagged data files we have around here. This is the gist of it altered for your file format. It is ugly but should get you well on your way.

I still think you should do this as a dlm but this exercise will be a good intro to IDL. Note that I have left a lot out and I make no claims that this is the way anyone should do this.

Enjoy.

-Rick

```

datasets = "
nsets = 0

;read in the batch file
openr, lun, filename, /get_lun

while not eof(lun) do begin
    line = "
    readf, lun, line

    ;remove comments and whitespace
    temp = strsplit(line, '//', /extract)
    line = strtrim(temp[0],2)

    ;check if this is a datablock tag
    dblock = strsplit(line, ':', /extract)

    if (N_ELEMENTS(dblock eq 1) then begin
        ;not a datablock tag
        case line of

            '[System]':begin

                while (line ne '[End]') do begin

                    line="
                    readf, lun, line

                    ;remove comments and whitespace
                    temp = strsplit(line, '//', /extract)
                    line = strtrim(temp[0],2)

                    ;parse the line of text for the keyword/param pair
                    kwpair = strsplit(strcompress(line, /remove_all), $
                        '=', /extract)
                    keyword = kwpair[0]

                    case keyword of
                        'Name':name = FIX(kwpair[1])
                        else: ;skip everything else
                    endcase
                endwhile
            end

            '[Data Sets]':begin
                while (line ne '[End]') do begin

                    line="

```

```

readf, lun, line

;remove comments and whitespace
line = strtrim((strsplit(line, '//', /extract))[0],2)

datasets = [datasets,line]
nsets = nsets + 1
endwhile
end

'[File Attributes]':begin

while (line ne '[End]') do begin

    line=""
    readf, lun, line

    ;remove comments and whitespace
    line = strtrim((strsplit(line, '//', /extract))[0],2)

    case line of
        '[Conditions File]':begin
            while (line ne '[End]') do begin
                line=""
                readf, lun, line

                ;remove comments and whitespace
                line = strtrim((strsplit(line, '//', $
                    /extract))[0],2)

                case line of
                    '[Properties]':begin
                        while (line ne '[End]') do begin
                            line=""
                            readf, lun, line

                            ;remove comments and whitespace
                            line = strtrim((strsplit(line,
'//', $

                                /extract))[0],2)

                            ;process keyword/data pairs

                        endwhile
                    end
                '[Frame Information]':begin

            end
        end
    end
end

```

```
        endcase
      endwhile
    end
    '[Saved File]':begin

    end
    else;;skip everything else

    endcase
  endwhile
end

else;;ignore everything else
endcase

endif else begin
  ;This is a datablock tag

  ;dblock[1] will be the dataset label

endelse

endwhile

free_lun, lun
```
