

---

Subject: Re: Defining constants in IDL?

Posted by [Dominic Zarro](#) on Sat, 22 Jun 2002 02:18:14 GMT

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

---

IDL is very forgiving. Just define a constant to be any name and set its value. For example,

```
IDL> a = 100
```

Just be careful not to redefine and, hence, overwrite the variable 'a'.

---

---

Subject: Re: Defining constants in IDL?

Posted by [mperrin+news](#) on Sat, 22 Jun 2002 02:31:41 GMT

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

---

Dominic Zarro <[dzarro@yahoo.com](mailto:dzarro@yahoo.com)> wrote:

```
> IDL is very forgiving. Just define a constant to be any name and set its
> value. For example,
>
> IDL> a = 100
>
> Just be careful not to redefine and, hence, overwrite the variable 'a'.
```

The problem with this approach is that it is only visible within one particular file. There's no mechanism akin to C's header files for getting a constant into multiple source files, something I frequently find myself missing. You can work around this using common blocks, but that's has its own problems. Perhaps the best solution is user-defined system variables but even that seems rather kludgy.

- Marshall

---

---

Subject: Re: Defining constants in IDL?

Posted by [ronn](#) on Sat, 22 Jun 2002 16:55:33 GMT

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

---

in article [af0nid\\$1ets\\$1@agate.berkeley.edu](#), Marshall Perrin at [mperrin+news@arkham.berkeley.edu](mailto:mperrin+news@arkham.berkeley.edu) wrote on 6/21/02 10:31 PM:

```
> Dominic Zarro <dzarro@yahoo.com> wrote:
>> IDL is very forgiving. Just define a constant to be any name and set its
>> value. For example,
>>
>> IDL> a = 100
>>
```

>> Just be careful not to redefine and, hence, overwrite the variable 'a'.  
>  
> The problem with this approach is that it is only visible within one  
> particular file. There's no mechanism akin to C's header files for  
> getting a constant into multiple source files, something I frequently find  
> myself missing.

Hi Marshall,

Actually there is something similar to including a header file. What you need is a file with no header (i.e. pro or function as the first line) and no end statement. For example,

```
a = 0
True = 1
False = 0
yes = 'YES'
```

could be saved in a file called constants.pro.

To include this in your own code you would do this.

```
pro myProcedure, p1, p2
```

```
@constants
```

```
...
...
```

```
end
```

The @ sign acts as an include statement to the compiler.

-Ronn

--

Ronn Kling

KRS, inc.

email: [ronn@rlkling.com](mailto:ronn@rlkling.com)

"Application Development with IDL" 1½ programming book updated for IDL5.5!

"Calling C from IDL, Using DLM's to extend your IDL code"

<http://www.rlkling.com/>

---

Subject: Re: Defining constants in IDL?

Posted by [Robert Stockwell](#) on Mon, 24 Jun 2002 13:04:36 GMT

Bob wrote:

> I have a quick question. I am a big newbie when it comes to IDL, but  
> the place that I work at uses it extensively. I am used to C/C++, and  
> I want to know how does one define a constant in IDL (similar to  
> #DEFINE TRUE 1 in C/C++). I have read through the user manuals, but to  
> no avail. Also, are there any good tutorials online to help a  
> struggling newbie like me? The user manuals 'cover' all the material,  
> but rather poorly.  
>

in your startup file, you can define system variables

```
DEFSYSV, 'imag', complex(0,1)  
DEFSYSV, 'twopi', 2*!dpi
```

```
print,!twopi
```

CHeers,  
bob

---

Subject: Re: Defining constants in IDL?  
Posted by [R.Bauer](#) on Mon, 24 Jun 2002 13:43:24 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Bob wrote:

>  
> I have a quick question. I am a big newbie when it comes to IDL, but  
> the place that I work at uses it extensively. I am used to C/C++, and  
> I want to know how does one define a constant in IDL (similar to  
> #DEFINE TRUE 1 in C/C++). I have read through the user manuals, but to  
> no avail. Also, are there any good tutorials online to help a  
> struggling newbie like me? The user manuals 'cover' all the material,  
> but rather poorly.

Dear Bob

there was a while ago someone else who likes to know something like this.

The whole thread on google you'll find:

<http://groups.google.com/groups?hl=en&lr=&ie=UTF-8&am;frame=right&th=a988307b311ea495&seekm=3BB9E19F.BB.8321E5%40fz-juelich.de#link1>

Reimar

--

Reimar Bauer

Institut fuer Stratosphaerische Chemie (ICG-I)  
Forschungszentrum Juelich  
email: R.Bauer@fz-juelich.de

-----  
a IDL library at ForschungsZentrum Juelich  
[http://www.fz-juelich.de/icg/icg1/idl\\_icglib/idl\\_lib\\_intro.h tml](http://www.fz-juelich.de/icg/icg1/idl_icglib/idl_lib_intro.h tml)  
=====

---

Subject: Re: Defining constants in IDL?  
Posted by [jeyadev](#) on Tue, 25 Jun 2002 17:37:28 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

In article <B93A247B.5660%ronn@rlkling.com>,  
ronn kling <ronn@rlkling.com> wrote:

>  
> Hi Marshall,  
>  
> Actually there is something similiar to including a header file. What you  
> need is a file with no header (i.e. pro or function as the first line) and  
> no end statement. For example,  
>  
> a = 0  
> True = 1  
> False = 0  
> yes = 'YES'  
>  
> could be saved in a file called constants.pro.  
>  
> To include this in your own code you would do this.  
>  
> pro myProcedure, p1, p2  
>  
> @constants  
>  
> ...  
> ...  
>  
> end

>

> The @ sign acts as an include statement to the compiler.

Alas, it fails with PV-Wave. Does anyone have a workaround?

```
kaveri 193% cat junk.pro
```

```
a = 1.0
```

```
b = 2.0
```

```
c = 3.0
```

```
kaveri 194% cat try.pro
```

```
pro try, x
```

```
@ junk
```

```
  print, x + a
```

```
  print, x + b
```

```
  print, x + c
```

```
end
```

```
.....
```

```
.....
```

```
WAVE> .run try
```

```
% Input line is too long for input buffer of 511 characters.
```

```
% Compiled module: TRY.
```

```
WAVE>
```

Not good, eh? Thought it was my file try.pro that was the problem.  
Turns out that it is not. Commenting the line out makes the file  
'compile'.

I must say that the lack of this facility is very irksome. It is  
one of the great advantages of Python, which, though similar to  
IDL/Wave has distinct advantages with namespaces, etc. There are  
many times that I would like to call a Wave "main" programme  
from another (just as I would Unix scripts) and not being able  
means doing some of the work with Python, etc.

--

Surendar Jeyadev      jeyadev@wrc.xerox.bounceback.com

Remove 'bounceback' for email address

---

---

Subject: Re: Defining constants in IDL?

Posted by [thompson](#) on Thu, 27 Jun 2002 22:51:09 GMT

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

---

I believe that the @ statement also works in PVWave. However, you have to be careful with @. You can't have any leading spaces, or spaces between it and the filename. So, you should change "@ junk" to "@junk", and make sure that the @ is all the way over to the left.

William Thompson

jeyadev@wrc.xerox.bounceback.com (Surendar Jeyadev) writes:

```
> In article <B93A247B.5660%ronn@rlkling.com>,
> ronn kling <ronn@rlkling.com> wrote:
>>
>> Hi Marshall,
>>
>> Actually there is something similiar to including a header file. What you
>> need is a file with no header (i.e. pro or function as the first line) and
>> no end statement. For example,
>>
>> a = 0
>> True = 1
>> False = 0
>> yes = 'YES'
>>
>> could be saved in a file called constants.pro.
>>
>> To include this in your own code you would do this.
>>
>> pro myProcedure, p1, p2
>>
>> @constants
>>
>> ...
>> ...
>>
>> end
>>
>> The @ sign acts as an include statement to the compiler.

> Alas, it fails with PV-Wave. Does anyone have a workaround?

> kaveri 193% cat junk.pro
> a = 1.0
> b = 2.0
> c = 3.0
```

> kaveri 194% cat try.pro

> pro try, x

> @ junk

> print, x + a

> print, x + b

> print, x + c

> end

>

> .....

> .....

> WAVE> .run try

> % Input line is too long for input buffer of 511 characters.

> % Compiled module: TRY.

> WAVE>

> Not good, eh? Thought it was my file try.pro that was the problem.

> Turns out that it is not. Commenting the line out makes the file

> 'compile'.

> I must say that the lack of this facility is very irksome. It is

> one of the great advantages of Python, which, though similar to

> IDL/Wave has distinct advantages with namespaces, etc. There are

> many times that I would like to call a Wave "main" programme

> from another (just as I would Unix scripts) and not being able

> means doing some of the work with Python, etc.

> --

> Surendar Jeyadev        jeyadev@wrc.xerox.bounceback.com

>

>                    Remove 'bounceback' for email address

---

Subject: Re: Defining constants in IDL?

Posted by [jeyadev](#) on Fri, 28 Jun 2002 22:51:38 GMT

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

---

In article <afg4st\$hrf\$1@skates.gsfc.nasa.gov>,  
William Thompson <thompson@orpheus.nascom.nasa.gov> wrote:

> I believe that the @ statement also works in PVWave. However, you have to be

> careful with @. You can't have any leading spaces, or spaces between it and

> the filename. So, you should change "@ junk" to "@junk", and make sure that  
> the @ is all the way over to the left.

Thanks. Will give it a shot on Monday. Wonder why there isn't a simple  
"include" command!

--

Surendar Jeyadev      jeyadev@wrc.xerox.bounceback.com

Remove 'bounceback' for email address

---