
Subject: Assigning command line arguments to variables in another procedure

Posted by [tryanblogs](#) on Sun, 10 Aug 2014 22:57:40 GMT

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

IDL beginner here! Let's say I have two procedures, PRO1 and PRO2. If I receive a command line argument in PRO2, how can I assign the argument value to a variable in PRO1?

Is this possible to accomplish without passing a parameter in a procedure call? Example:

```
PRO PRO2
value = command_line_argument
PRO1, value
END
```

```
PRO PRO1, value
A = value
END
```

I have previously tried to make an object reference , 'My', to PRO1, but I receive a syntax error.

```
PRO PRO2
opts = ob_new('mg_options)
opts.addOption, 'value', 'v'
opts.parseArgs,error_message = errorMsg
```

```
My = obj_new('PRO1')
My.A=opts.get('value') ;syntax error on this line
END
```

Any insight is very much appreciated!

Subject: Re: Assigning command line arguments to variables in another procedure

Posted by [Matthew Argall](#) on Sun, 10 Aug 2014 23:12:53 GMT

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

How about this?

```
IDL> myPro1, 'This is my argument.'
  In myPro1
  In myPro2
  This is my argument.
```

```
-----
pro myPro2, value
  print, 'In myPro2'
  print, value
```

end

```
pro myPro1, input_from_command_line
  print, 'In myPro1'
  myPro2, input_from_command_line
end
```

Subject: Re: Assigning command line arguments to variables in another procedure
Posted by [tryanblogs](#) on Mon, 11 Aug 2014 02:32:44 GMT

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

Matthew,

Thank you for your response. I'm not sure this will perform the desired action. I apologize if I'm missing something obvious, but passing values in a procedure call doesn't seem to accomplish what I'm trying to do. In the program I am attempting to write, the variables that will be assigned values from command line arguments are not predetermined and will vary with each case, so I don't think passing values through procedure arguments will work. Is it possible to assign variables in Pro1 with command line argument values from Pro2 using object references?

In java, it would look something like this:

```
Class Pro2 {
  Pro1 My = new Pro1()
  My.Pro1variable = command_line_args
}
```

where My is the object reference of Pro1 and the Pro1variable is given the value of the command line inputted in Pro2.

Subject: Re: Assigning command line arguments to variables in another procedure
Posted by [Yngvar Larsen](#) on Mon, 11 Aug 2014 09:56:10 GMT

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

"You are doing it all wrong". You should read about classes in IDL:

http://www.exelisvis.com/docs/ref_part2.html

http://www.exelisvis.com/docs/creating_custom_objects.html

Hint #1: Unlike in Java, not everything is a class in IDL.

Hint #2: PRO does not implement a class, but a procedure ("function" with no return value). Thus "My = obj_new('PRO1')" with PRO1 being a procedure, does not make sense.

Hint #3: Even if PRO1 was a class with a member called A, you cannot assign to the member

with "My.A" in IDL.

On Monday, 11 August 2014 00:57:40 UTC+2, tryan...@gmail.com wrote:

> IDL beginner here! Let's say I have two procedures, PRO1 and PRO2. If I receive a command line argument in PRO2, how can I assign the argument value to a variable in PRO1?

>
>
>
> Is this possible to accomplish without passing a parameter in a procedure call? Example:

>
>
> PRO PRO2
>
> value = command_line_argument
>
> PRO1, value
>
> END

>
>
>
> PRO PRO1, value
>
> A = value
>
> END

>
>
>
> I have previously tried to make an object reference, 'My', to PRO1, but I receive a syntax error.

>
>
>
> PRO PRO2
>
> opts = ob_new('mg_options')
>
> opts.addOption, 'value', 'v'
>
> opts.parseArgs,error_message = errorMsg
>
>
>
> My = obj_new('PRO1')
>
> My.A=opts.get('value') ;syntax error on this line
>

> END
>
>
>
> Any insight is very much appreciated!

Subject: Re: Assigning command line arguments to variables in another procedure
Posted by skymaxwell@gmail.com on Mon, 11 Aug 2014 15:19:06 GMT
[View Forum Message](#) <> [Reply to Message](#)

Look at IDL help for `_EXTRA` and `_REF_EXTRA` keywords
