Subject: Re: Procedures with variable/dynamic argument list??
Posted by Gray on Fri, 22 Oct 2010 08:57:59 GMT
View Forum Message <> Reply to Message

On Oct 21, 9:49 pm, SonicKenking <ywa...@gmail.com> wrote:
> Is it possible in IDL to code a procedure using variable/dynamic
> argument list?? i.e. the number of arguments is arbitrary and the
> procedure is able to accept them all with some argument array?
>
> Similar to the "va_list" in C or "*args" in Python.
>
> The Print command in IDL is a good example too for accepting arbitrary
> number of arguments and print them all. Unfortunately, the Print
> command is not written in IDL, hence no sneak peek of the source code
> ~~

You want the _EXTRA keyword.

Subject: Re: Procedures with variable/dynamic argument list??
Posted by SonicKenking on Fri, 22 Oct 2010 09:04:46 GMT
View Forum Message <> Reply to Message

On Oct 22, 7:57 pm, Gray <grayliketheco...@gmail.com> wrote:
> On Oct 21, 9:49 pm, SonicKenking <ywa...@gmail.com> wrote:
>
>> Is it possible in IDL to code a procedure using variable/dynamic
>> argument list?? i.e. the number of arguments is arbitrary and the
>> procedure is able to accept them all with some argument array?
>
>> Similar to the "va_list" in C or "*args" in Python.
>
>> The Print command in IDL is a good example too for accepting arbitrary
>> number of arguments and print them all. Unfortunately, the Print
>> command is not written in IDL, hence no sneak peek of the source code
>> ~~
>
> You want the _EXTRA keyword.

hmm, correct me if I am wrong. But _EXTRA is only good for accepting
extra keyword arguments, not positional arguments. I need something
similar to _Extra, but accept positional arguments.

Subject: Re: Procedures with variable/dynamic argument list??
Posted by Kenneth P. Bowman on Fri, 22 Oct 2010 13:56:00 GMT
View Forum Message <> Reply to Message

In article
<d5d024b5-fb3e-438e-9277-c4dc032c35e2@n24g2000prj.googlegroups.com>,
SonicKenking <ywangd@gmail.com> wrote:

> Is it possible in IDL to code a procedure using variable/dynamic
> argument list?? i.e. the number of arguments is arbitrary and the
> procedure is able to accept them all with some argument array?
>
> Similar to the "va_list" in C or "*args" in Python.
>
> The Print command in IDL is a good example too for accepting arbitrary
> number of arguments and print them all. Unfortunately, the Print
> command is not written in IDL, hence no sneak peek of the source code
> ~~

IDL procedures and functions are flexible about how many
arguments you provide, as long as you don't pass too many.

You do have to provide at least as many dummy arguments
in the procedure definition as you intend to use. That is

```
   PRO VAR_ARGS, a1, a2, a3, a4
```

can be called as

```
   VAR_ARGS, x, y
```

It is up to the procedure to know what to do in that
case. (See the N_PARAMS function.)

As far as I know, there is no direct mechanism for
handling an arbitrary number of arguments other than
making a list in the procedure/function definition.

Ken

Subject: Re: Procedures with variable/dynamic argument list??
Posted by pgrigis on Fri, 22 Oct 2010 14:26:38 GMT
View Forum Message <> Reply to Message

On Oct 22, 9:56 am, "Kenneth P. Bowman" <k-bow...@null.edu> wrote:
> In article
> < d5d024b5-fb3e-438e-9277-c4dc032c3...@n24g2000prj.googlegroup s.com >,
>
>   SonicKenking <ywa...@gmail.com> wrote:
>> Is it possible in IDL to code a procedure using variable/dynamic
>> argument list?? i.e. the number of arguments is arbitrary and the

>> procedure is able to accept them all with some argument array?
>
>> Similar to the "va_list" in C or "*args" in Python.
>
>> The Print command in IDL is a good example too for accepting arbitrary
>> number of arguments and print them all. Unfortunately, the Print
>> command is not written in IDL, hence no sneak peek of the source code
>> ~~
>
> IDL procedures and functions are flexible about how many
> arguments you provide, as long as you don't pass too many.
>
> You do have to provide at least as many dummy arguments
> in the procedure definition as you intend to use. That is
>
>   PRO VAR_ARGS, a1, a2, a3, a4
>
> can be called as
>
>   VAR_ARGS, x, y
>
> It is up to the procedure to know what to do in that
> case. (See the N_PARAMS function.)
>
> As far as I know, there is no direct mechanism for
> handling an arbitrary number of arguments other than
> making a list in the procedure/function definition.
>
> Ken

It comes out pretty ugly indeed... here my "pritn" function
that I use to work around my bad spelling (!)
I guess I didn't know about n_params() when I wrote that.

Ciao,
Paolo

```
PRO pritn,a1,a2,a3,_extra=_extra

IF exist(a3) THEN print,a1,a2,a3,_extra=_extra ELSE $
   IF exist(a2) THEN print,a1,a2,_extra=_extra ELSE $
      IF exist(a1) THEN print,a1,_extra=_extra

END

;from solarsoft
function exist,var
```

```
return,n_elements(var) ne 0

end
```

Subject: Re: Procedures with variable/dynamic argument list??
Posted by chris_torrence@NOSPAM on Fri, 22 Oct 2010 15:14:47 GMT
View Forum Message <> Reply to Message

On Oct 22, 8:26 am, Paolo <pgri...@gmail.com> wrote:

```
>
>  It comes out pretty ugly indeed... here my "pritn" function
>  that I use to work around my bad spelling (!)
>  I guess I didn't know about n_params() when I wrote that.
>
>  Ciao,
>  Paolo
>
>  PRO pritn,a1,a2,a3,_extra=_extra
>
>  IF exist(a3) THEN print,a1,a2,a3,_extra=_extra ELSE $
>     IF exist(a2) THEN print,a1,a2,_extra=_extra ELSE $
>        IF exist(a1) THEN print,a1,_extra=_extra
>
>  END
>
>  ;from solarsoft
>  function exist,var
>
>  return,n_elements(var) ne 0
>
>  end
```

Hi all,

You might look at SCOPE_VARFETCH. You could dynamically construct the argument name, and then pass the string to SCOPE_VARFETCH to retrieve the value. Something like:

```
PRO pritn,a1,a2,a3,_extra=_extra
  for i=1,n_params() do print, SCOPE_VARFETCH('a'+strtrim(i,
2)),_extra=_extra
end
```

This only helps in cases where you can break up the arguments and do a call with each one separately.

Paolo, if I could suggest a tweak, you could use a CASE statement
instead:

```
case n_params() of
1: print,a1,_extra=_extra
2: print,a1,a2,_extra=_extra
3: print,a1,a2,a3,_extra=_extra
else: message,'Incorrect number of arguments'
endcase
```

This avoids all of those nasty if/else blocks.

-Chris
ITTVIS

Subject: Re: Procedures with variable/dynamic argument list??

Posted by pgrigis on Fri, 22 Oct 2010 15:50:14 GMT

View Forum Message <> Reply to Message

On Oct 22, 11:14 am, Chris Torrence <gorth...@gmail.com> wrote:

```
>  On Oct 22, 8:26 am, Paolo <pgri...@gmail.com> wrote:
>
>
>
>
>
>>  It comes out pretty ugly indeed... here my "pritn" function
>>  that I use to work around my bad spelling (!)
>>  I guess I didn't know about n_params() when I wrote that.
>
>>  Ciao,
>>  Paolo
>
>>  PRO pritn,a1,a2,a3,_extra=_extra
>
>>  IF exist(a3) THEN print,a1,a2,a3,_extra=_extra ELSE $
>>     IF exist(a2) THEN print,a1,a2,_extra=_extra ELSE $
>>        IF exist(a1) THEN print,a1,_extra=_extra
>
>>  END
>
>>  ;from solarsoft
>>  function exist,var
>
>>  return,n_elements(var) ne 0
>
>>  end
```

>
> Hi all,
>
> You might look at SCOPE_VARFETCH. You could dynamically construct the
> argument name, and then pass the string to SCOPE_VARFETCH to retrieve
> the value. Something like:
>
> PRO pritn,a1,a2,a3,_extra=_extra
>   for i=1,n_params() do print, SCOPE_VARFETCH('a'+strtrim(i,
> 2)),_extra=_extra
> end
>
> This only helps in cases where you can break up the arguments and do a
> call with each one separately.
>
> Paolo, if I could suggest a tweak, you could use a CASE statement
> instead:
>
> case n_params() of
> 1: print,a1,_extra=_extra
> 2: print,a1,a2,_extra=_extra
> 3: print,a1,a2,a3,_extra=_extra
> else: message,'Incorrect number of arguments'
> endcase
>

Yes, the CASE version is nicer and that's how I would have
wrote it if I had known about n_params() back then... :)
You always learn something new :)

Thanks,
Paolo

> This avoids all of those nasty if/else blocks.
>
> -Chris
> ITTVIS

Subject: Re: Procedures with variable/dynamic argument list??
Posted by SonicKenking on Mon, 25 Oct 2010 07:32:17 GMT

View Forum Message <> Reply to Message

Thanks all for the reply.
It is good to be clear that it is not directly possible to make a
dynamic argument list. I'll just go with the dummy parameter way and
use scope_varfetch to loop on the actual passed arguments.

Subject: Re: Procedures with variable/dynamic argument list??
Posted by Michael Galloy on Mon, 25 Oct 2010 13:51:54 GMT

View Forum Message <> Reply to Message

On 10/25/10 1:32 am, SonicKenking wrote:

> Thanks all for the reply.
> It is good to be clear that it is not directly possible to make a
> dynamic argument list. I'll just go with the dummy parameter way and
> use scope_varfetch to loop on the actual passed arguments.

You can do it from within a DLM routine, though.

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