Subject: Re: Starting a for loop within an if loop
Posted by David Fanning on Thu, 20 Dec 2007 04:14:48 GMT
View Forum Message <> Reply to Message

Mat Smith writes:

> I'm trying to write a program with some keywords, so that depending
> upon my mood I can alter what it does.
>
> What I want to do is most of the time just run the program, but I want
> to include an option to do a for loop.
>
> Basically, I need something like:
>
> IF keyword_set(mc) THEN BEGIN
> FOR i=0,n-1 DO BEGIN
> t=d[i]
> ENDIF
>
> and later in the program
>
> IF keyword_set(mc) THEN BEGIN
> ENDFOR
> ENDIF
>
> Thus the rest of the program runs normally with and without the for
> loop (it just uses t).
> However, IDL doesn't understand this - it wants an ENDFOR before the
> ENDIF.
>
> Any thoughts on how I can get around this? It's entirely possible that
> I'm missing something obvious.

It's possible. Have you taught yourself to program?
Sometimes there can be gaps. :-)

Anyway, I think I would do something like this:

```
IF keyword_set(mc) THEN endloop = n ELSE endloop = 1
FOR j=0,endloop-1 DO BEGIN
 ...
ENDFOR
```

This way, if your keyword is set, you will do the loop n times,
otherwise you will do the loop just once.

Cheers,

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

Subject: Re: Starting a for loop within an if loop
Posted by Mat Smith on Thu, 20 Dec 2007 06:15:21 GMT
View Forum Message <> Reply to Message

On Dec 20, 1:14 am, David Fanning <n...@dfanning.com> wrote:
> Mat Smith writes:
>> I'm trying to write a program with some keywords, so that depending
>> upon my mood I can alter what it does.
>
>> What I want to do is most of the time just run the program, but I want
>> to include an option to do a for loop.
>
>> Basically, I need something like:
>
>> IF keyword_set(mc) THEN BEGIN
>> FOR i=0,n-1 DO BEGIN
>> t=d[i]
>> ENDIF
>
>> and later in the program
>
>> IF keyword_set(mc) THEN BEGIN
>> ENDFOR
>> ENDIF
>
>> Thus the rest of the program runs normally with and without the for
>> loop (it just uses t).
>> However, IDL doesn't understand this - it wants an ENDFOR before the
>> ENDIF.
>
>> Any thoughts on how I can get around this? It's entirely possible that
>> I'm missing something obvious.
>
> It's possible. Have you taught yourself to program?
> Sometimes there can be gaps. :-)
>
> Anyway, I think I would do something like this:
>
> IF keyword_set(mc) THEN endloop = n ELSE endloop = 1
> FOR j=0,endloop-1 DO BEGIN

> ...
> ENDFOR
>
> This way, if your keyword is set, you will do the loop n times,
> otherwise you will do the loop just once.
>
> Cheers,
>
> David
> --
> David Fanning, Ph.D.
> Fanning Software Consulting, Inc.
> Coyote's Guide to IDL Programming:http://www.dfanning.com/
> Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Hi David,

I have indeed taught myself to program, and there are undoubtedly huge gaps!! I've never been one for learn the basics - I'm more of a dive in and solve problems as they come!

If I read your reply correctly, I don't think that it'll do what I want it to, but I can adapt it so that it will!

The problem was that I wanted to have different structures depending on if the keyword is set.

That is if the keyword isn't set then it will have a particular values, but if it is set, then it will run the loop AND the values will always be different to when the keyword isn't set (i.e. loop of 1 isn't the not keyword).

So before I had

```
IF NOT keyword_set(mc) THEN BEGIN
t=s
ENDIF ELSE BEGIN
FOR i=0,k-1 DO BEGIN
t=d[i]
ENDELSE
... working on t
ENDFOR - which obviously didn't work
```

But I can adapt your suggestion to make it work - i.e. in the for loop call another if loop! e.g.

```
IF keyword_set(mc) THEN endloop = n ELSE endloop = 1
FOR j=0,endloop-1 DO BEGIN
```

IF NOT keyword_set(mc) THEN BEGIN
t=s[j]
ENDIF ELSE BEGIN
t=d[j]
ENDELSE
...woking on t
ENDFOR

Does this make sense? It's very long-winded (there must be a quicker way!). I'm trying to be clever and I'm probably making my programs too generic. I'm also probably missing something simple (thanks to the learning it as I go!)

Thanks

Mat

Subject: Re: Starting a for loop within an if loop
Posted by Mat Smith on Thu, 20 Dec 2007 06:15:35 GMT
View Forum Message <> Reply to Message

On Dec 20, 1:14 am, David Fanning <n...@dfanning.com> wrote:
> Mat Smith writes:
>> I'm trying to write a program with some keywords, so that depending
>> upon my mood I can alter what it does.
>
>> What I want to do is most of the time just run the program, but I want
>> to include an option to do a for loop.
>
>> Basically, I need something like:
>
>> IF keyword_set(mc) THEN BEGIN
>> FOR i=0,n-1 DO BEGIN
>> t=d[i]
>> ENDIF
>
>> and later in the program
>
>> IF keyword_set(mc) THEN BEGIN
>> ENDFOR
>> ENDIF
>
>> Thus the rest of the program runs normally with and without the for
>> loop (it just uses t).
>> However, IDL doesn't understand this - it wants an ENDFOR before the
>> ENDIF.
>

>> Any thoughts on how I can get around this? It's entirely possible that
>> I'm missing something obvious.
>
> It's possible. Have you taught yourself to program?
> Sometimes there can be gaps. :-)
>
> Anyway, I think I would do something like this:
>
> IF keyword_set(mc) THEN endloop = n ELSE endloop = 1
> FOR j=0,endloop-1 DO BEGIN
> ...
> ENDFOR
>
> This way, if your keyword is set, you will do the loop n times,
> otherwise you will do the loop just once.
>
> Cheers,
>
> David
> --
> David Fanning, Ph.D.
> Fanning Software Consulting, Inc.
> Coyote's Guide to IDL Programming:http://www.dfanning.com/
> Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Hi David,

I have indeed taught myself to program, and there are undoubtedly huge gaps!! I've never been one for learn the basics - I'm more of a dive in and solve problems as they come!

If I read your reply correctly, I don't think that it'll do what I want it to, but I can adapt it so that it will!

The problem was that I wanted to have different structures depending on if the keyword is set.

That is if the keyword isn't set then it will have a particular values, but if it is set, then it will run the loop AND the values will always be different to when the keyword isn't set (i.e. loop of 1 isn't the not keyword).

So before I had

```
IF NOT keyword_set(mc) THEN BEGIN
t=s
ENDIF ELSE BEGIN
FOR i=0,k-1 DO BEGIN
```

t=d[i]
ENDELSE
... working on t
ENDFOR - which obviously didn't work

But I can adapt your suggestion to make it work - i.e. in the for loop call another if loop! e.g.

IF keyword_set(mc) THEN endloop = n ELSE endloop = 1
FOR j=0,endloop-1 DO BEGIN
IF NOT keyword_set(mc) THEN BEGIN
t=s[j]
ENDIF ELSE BEGIN
t=d[j]
ENDELSE
...woking on t
ENDFOR

Does this make sense? It's very long-winded (there must be a quicker way!). I'm trying to be clever and I'm probably making my programs too generic. I'm also probably missing something simple (thanks to the learning it as I go!)

Thanks

Mat

Subject: Re: Starting a for loop within an if loop
Posted by Mat Smith on Thu, 20 Dec 2007 06:50:54 GMT
View Forum Message <> Reply to Message

On Dec 20, 1:14 am, David Fanning <n...@dfanning.com> wrote:
> Mat Smith writes:
>> I'm trying to write a program with some keywords, so that depending
>> upon my mood I can alter what it does.
>
>> What I want to do is most of the time just run the program, but I want
>> to include an option to do a for loop.
>
>> Basically, I need something like:
>
>> IF keyword_set(mc) THEN BEGIN
>> FOR i=0,n-1 DO BEGIN
>> t=d[i]
>> ENDIF
>
>> and later in the program

>
>> IF keyword_set(mc) THEN BEGIN
>> ENDFOR
>> ENDIF
>
>> Thus the rest of the program runs normally with and without the for
>> loop (it just uses t).
>> However, IDL doesn't understand this - it wants an ENDFOR before the
>> ENDIF.
>
>> Any thoughts on how I can get around this? It's entirely possible that
>> I'm missing something obvious.
>
> It's possible. Have you taught yourself to program?
> Sometimes there can be gaps. :-)
>
> Anyway, I think I would do something like this:
>
> IF keyword_set(mc) THEN endloop = n ELSE endloop = 1
> FOR j=0,endloop-1 DO BEGIN
> ...
> ENDFOR
>
> This way, if your keyword is set, you will do the loop n times,
> otherwise you will do the loop just once.
>
> Cheers,
>
> David
> --
> David Fanning, Ph.D.
> Fanning Software Consulting, Inc.
> Coyote's Guide to IDL Programming:http://www.dfanning.com/
> Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Hi David,

I have indeed taught myself to program, and there are undoubtedly huge
gaps!! I've never been one for learn the basics - I'm more of a dive
in and solve problems as they come!

If I read your reply correctly, I don't think that it'll do what I
want it to, but I can adapt it so that it will!

The problem was that I wanted to have different structures depending
on if the keyword is set.

That is if the keyword isn't set then it will have a particular

values, but if it is set, then it will run the loop AND the values
will always be different to when the keyword isn't set (i.e. loop of 1
isn't the not keyword).

So before I had

IF NOT keyword_set(mc) THEN BEGIN
t=s
ENDIF ELSE BEGIN
FOR i=0,k-1 DO BEGIN
t=d[i]
ENDELSE
... working on t
ENDFOR - which obviously didn't work

But I can adapt your suggestion to make it work - i.e. in the for loop
call another if loop! e.g.

IF keyword_set(mc) THEN endloop = n ELSE endloop = 1
FOR j=0,endloop-1 DO BEGIN
IF NOT keyword_set(mc) THEN BEGIN
t=s[j]
ENDIF ELSE BEGIN
t=d[j]
ENDELSE
...woking on t
ENDFOR

Does this make sense? It's very long-winded (there must be a quicker
way!). I'm trying to be clever and I'm probably making my programs too
generic. I'm also probably missing something simple (thanks to the
learning it as I go!)

Thanks

Mat

Subject: Re: Starting a for loop within an if loop
Posted by Allan Whiteford on Thu, 20 Dec 2007 09:31:40 GMT
View Forum Message <> Reply to Message

Mat Smith wrote:
> On Dec 20, 1:14 am, David Fanning <n...@dfanning.com> wrote:
>
>> Mat Smith writes:
>>
>>> I'm trying to write a program with some keywords, so that depending

>>> upon my mood I can alter what it does.
>>
>>> What I want to do is most of the time just run the program, but I want
>>> to include an option to do a for loop.
>>
>>> Basically, I need something like:
>>
>>> IF keyword_set(mc) THEN BEGIN
>>> FOR i=0,n-1 DO BEGIN
>>> t=d[i]
>>> ENDIF
>>
>>> and later in the program
>>
>>> IF keyword_set(mc) THEN BEGIN
>>> ENDFOR
>>> ENDIF
>>
>>> Thus the rest of the program runs normally with and without the for
>>> loop (it just uses t).
>>> However, IDL doesn't understand this - it wants an ENDFOR before the
>>> ENDIF.
>>
>>> Any thoughts on how I can get around this? It's entirely possible that
>>> I'm missing something obvious.
>>
>> It's possible. Have you taught yourself to program?
>> Sometimes there can be gaps. :-)
>>
>> Anyway, I think I would do something like this:
>>
>>   IF keyword_set(mc) THEN endloop = n ELSE endloop = 1
>>   FOR j=0,endloop-1 DO BEGIN
>>   ...
>>   ENDFOR
>>
>> This way, if your keyword is set, you will do the loop n times,
>> otherwise you will do the loop just once.
>>
>> Cheers,
>>
>> David
>> --
>> David Fanning, Ph.D.
>> Fanning Software Consulting, Inc.
>> Coyote's Guide to IDL Programming:http://www.dfanning.com/
>> Sepore ma de ni thui. ("Perhaps thou speakest truth.")
>

>
> Hi David,
>
> I have indeed taught myself to program, and there are undoubtedly huge
> gaps!! I've never been one for learn the basics - I'm more of a dive
> in and solve problems as they come!
>
> If I read your reply correctly, I don't think that it'll do what I
> want it to, but I can adapt it so that it will!
>
> The problem was that I wanted to have different structures depending
> on if the keyword is set.
>

You might want to look at making your "... working on t" a subroutine in
its own right, this would give you code like:

```
pro doit,t
 ... work on t
end

if keyword_set(mc) then begin
 for i=0,k-1 do begin
  doit,d[i]
 end
endif else begin
 doit,s
endelse
```

it will give you slightly more flexibility to put your core code inside
different structures etc. If mc stands for Monte Carlo then you almost
certainly want to separate your model from the controlling code.

I'd guess you'd also want to pass an answer back out of "doit" in which
case you might want to make it a function or add more parameters.

> That is if the keyword isn't set then it will have a particular
> values, but if it is set, then it will run the loop AND the values
> will always be different to when the keyword isn't set (i.e. loop of 1
> isn't the not keyword).
>
> So before I had
>
> IF NOT keyword_set(mc) THEN BEGIN
> t=s
> ENDIF ELSE BEGIN
> FOR i=0,k-1 DO BEGIN
> t=d[i]

> ENDELSE
> ... working on t
> ENDFOR - which obviously didn't work
>

It's not a completely unreasonable thing to want - it's just that IDL
won't let you do it.

> But I can adapt your suggestion to make it work - i.e. in the for loop
> call another if loop! e.g.
>
> IF keyword_set(mc) THEN endloop = n ELSE endloop = 1
> FOR j=0,endloop-1 DO BEGIN
> IF NOT keyword_set(mc) THEN BEGIN
> t=s[j]
> ENDIF ELSE BEGIN
> t=d[j]
> ENDELSE
> ...woking on t
> ENDFOR
>
> Does this make sense? It's very long-winded (there must be a quicker
> way!). I'm trying to be clever and I'm probably making my programs too
> generic. I'm also probably missing something simple (thanks to the
> learning it as I go!)

In this context, you'd be better with something like:

```
if keyword_set(mc) then begin
 endloop=n
 t=d ; d can be an array
endif else begin
 endloop=1
 t=s ; s can be a scalar
endelse

for j=0,endloop-1 do begin
 ; now work on t[i]
end
```

this will save doing an "if" inside of your "for". IDL will not optimise
this away for you like some other compilers might claim to.

Thanks,

Allan

Subject: Re: Starting a for loop within an if loop
Posted by Wox on Thu, 20 Dec 2007 10:08:34 GMT
View Forum Message <> Reply to Message

On Wed, 19 Dec 2007 19:57:50 -0800 (PST), Mat Smith
<mat.smith@port.ac.uk> wrote:

> IF keyword_set(mc) THEN BEGIN
> FOR i=0,n-1 DO BEGIN
> t=d[i]
> ENDIF
>
> and later in the program
>
> IF keyword_set(mc) THEN BEGIN
> ENDFOR
> ENDIF

You're thinking of preprocessor directives like in e.g. C++

#ifdef <token>
/* code */
#endif

So at compile time it looks whether <token> is defined or not and includes or excludes code in compilation. Even if IDL would be compiled and have a preprocessor, you only know whether <token> is defined ("keyword is set" if you will) at runtime.

Btw, it's an "if statement" not an "if loop". I don't want to offend you, but I was glad someone told me that ones. Time to return the favour ;-).

Subject: Re: Starting a for loop within an if loop
Posted by Carsten Lechte on Thu, 20 Dec 2007 11:19:26 GMT
View Forum Message <> Reply to Message

Mat Smith wrote:
> Basically, I need something like:
>
> IF keyword_set(mc) THEN BEGIN
> FOR i=0,n-1 DO BEGIN
> t=d[i]
> ENDIF
>
> and later in the program
>

> IF keyword_set(mc) THEN BEGIN
> ENDFOR
> ENDIF

Others have pointed out some solutions for your problem, so I
will comment on why your approach must fail.

The reason is: Computers do not understand programming languages.

Source code in IDL, C, perl etc. must be translated into bit
patterns (the "machine code") that the CPU of the computer can
use directly. The abstract concept of a FOR loop is translated
into a certain set of machine instructions and then executed.
Fro this to work, the loop has to be present in the source code
in its entirety.

In the source code, i.e. the stuff that the human writes, the
FOR loop is often represented by several units/lines of code:
The "FOR ...", the code that is to be looped over, and the
termination statement "ENDFOR". In most programming languages,
these are all obligatory components that have to be present at
the time that the code is translated into machine code (i.e.
during the compilation phase.)

What you tried was to make conditional statements that can only
be evaluated during run-time, i.e. after the compilation phase --
but the compiler does not get that far. Your only hope is to use
the FOR loop as a complete unit inside your conditional statements.
That way, the compiler can translate the whole loop, and during
run-time, the program can decide if the loop should be run or not.

I find it hard to express these concepts (other than saying
"compilers do not work like that.")

Note that in C, you actually can use the preprocessor to do
something like that, but this works because the conditionals
are decided before compilation, and the actual compiler either
sees the complete FOR loop, or none at all (or a loop with one
half missing, if you messed up the #ifdef's...)

chl

Subject: Re: Starting a for loop within an if loop
Posted by David Fanning on Thu, 20 Dec 2007 13:43:52 GMT
View Forum Message <> Reply to Message

Carsten Lechte writes:

> Others have pointed out some solutions for your problem, so I
> will comment on why your approach must fail.
>
> The reason is: Computers do not understand programming languages.

Hi Carsten,

I've submitted your name for Honorable Mention at
the next IEPA convention. I've put it in the category
Quirky Humor, Followed By Great Information. Nice post! :-)

Cheers,

David

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

Subject: Re: Starting a for loop within an if loop
Posted by edward.s.meinel@aero. on Thu, 20 Dec 2007 15:01:54 GMT
View Forum Message <> Reply to Message

On Dec 19, 10:57 pm, Mat Smith <mat.sm...@port.ac.uk> wrote:

> Any thoughts on how I can get around this? It's entirely possible that
> I'm missing something obvious.
> Thanks
> Mat

Thanks a lot, Mat. Now I have a headache.

My take on this is that you are treating IDL like C, C++, FORTRAN (pre-
F90), ... You managed to suck me (and apparently everyone else) into
the same thought pattern with that convoluted little piece of code.
Now that I've cleared my mind, why can't you just do:

```
CASE mc OF
  1:    t = d
  ELSE: t = s
ENDCASE
```

... working on t

Ed

Subject: Re: Starting a for loop within an if loop
Posted by Carsten Lechte on Fri, 21 Dec 2007 10:12:23 GMT
View Forum Message <> Reply to Message

David Fanning wrote:
> Nice post! :-)

Why, thank you <blush>

chl
