Subject: Re: Multiple values from a function?
Posted by meron on Tue, 04 Jul 2000 07:00:00 GMT
View Forum Message <> Reply to Message

In article <3961EFA8.49657FB6@mathstat.dal.ca>, Simon de Vet <simon@mathstat.dal.ca> writes:

> I have a function that does a whole lot (it started out as an
> independent program). It works very well when returning one of it's
> calculated arrays.
>
> It also produces a number of other arrays which I'd like to have access
> to.  Unfortunately, I have discovered that a function can only return
> one value.
>
> How can I get the other arrays out of it? The code is intertwined, so it
> would be hard to break into 2 or 3 independent functions.
>

You can easily return them using keyword parameters.  Lets say that your function FUN calculates 3 arrays (internally known as XARR, YARR and ZARR) and returns XARR.  Then, modify the definition of the function to something like

```
Fuction FUN, ...  ..., second = yarr, third = zarr

 ....

   return, xarr
end
```

Now, if you'll issue a call like

```
u = Fun( .... ,second = v, third = w)
```

then after the call U will contain the calculated XARR, V will contain YARR, w will contain ZARR.  This can be extended indefinitely.  Note that neither U not V or W need to be defined at all, prior to the call.

Mati Meron                        | "When you argue with a fool,
meron@cars.uchicago.edu         |  chances are he is doing just the same"

Subject: Re: Multiple values from a function?
Posted by Martin Schultz on Tue, 04 Jul 2000 07:00:00 GMT
View Forum Message <> Reply to Message

Simon de Vet wrote:
>

> I have a function that does a whole lot (it started out as an
> independent program). It works very well when returning one of it's
> calculated arrays.
>
> It also produces a number of other arrays which I'd like to have access
> to.  Unfortunately, I have discovered that a function can only return
> one value.
>
> How can I get the other arrays out of it? The code is intertwined, so it
> would be hard to break into 2 or 3 independent functions.
>
> Concatenation of arrays is not an option, since they are all of
> different sizes.  I can get around the problem by doing the calculation
> in the main level program instead of in a function, but that's a little
> ackward. Looking at the help, structures look like a possibility, but I
> don't understand how they work.
>
> Help!
>
> Simon

Second alternative: store all your results in a structure. Example:

```
function mything, in1, in2

  sum=in1+in2
  diff=in1-in2

  result = { sum:sum, diff:diff }
  return,result
end
```

You can then access the results as in:

```
  thunder = mything(5.,20.)
  print,'Sum is : ',thunder.sum,'   Difference is : ',thunder.diff
```

One addition to Craig's answer: the parameter could also be a keyword,
e.g.

```
function mything, in, out=out
```

This is particularily recommended if you judge the output as optional.
In this case
you can use the arg_present function to determine whether output shall
be computed or not:

```
  ...
  if arg_present(out) then begin
     ;; long-winded computation worth at least 3 trips to coffee maker
     ...
  endif
```

Cheers,
Martin
--

```
 [[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[ [[[[[[[
[[ Dr. Martin Schultz   Max-Planck-Institut fuer Meteorologie    [[
[[                      Bundesstr. 55, 20146 Hamburg             [[
[[                      phone: +49 40 41173-308                  [[
[[                      fax:   +49 40 41173-298                  [[
[[ martin.schultz@dkrz.de                                        [[
 [[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[ [[[[[[[
```

## Subject: Re: Multiple values from a function?
## Posted by Craig Markwardt on Tue, 04 Jul 2000 07:00:00 GMT

View Forum Message <> Reply to Message

Simon de Vet <simon@mathstat.dal.ca> writes:

> I have a function that does a whole lot (it started out as an
> independent program). It works very well when returning one of it's
> calculated arrays.
>
> It also produces a number of other arrays which I'd like to have access
> to.  Unfortunately, I have discovered that a function can only return
> one value.
>
> How can I get the other arrays out of it? The code is intertwined, so it
> would be hard to break into 2 or 3 independent functions.
> ...

The first and simple answer is, "no," you can't RETURN multiple values like you desire.

However, the second and more useful answer is that you CAN return values in positional or keyword parameters.  IDL is pass-by-reference in most cases, so you can simply set the parameter value to whatever you want.  I do this all the time to return subsidiary values, and it works for both procedures and functions.

```
function example, in, out
  out = in^2   ;; calling routine will have new value of OUT upon return
  return in^3
end
```

Good luck,
Craig

```
--
 ------------------------------------------------------------------------- --------------
Craig B. Markwardt, Ph.D.         EMAIL:    craigmnet@cow.physics.wisc.edu
Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response
 ------------------------------------------------------------------------- --------------
```

Subject: Re: Multiple values from a function?
Posted by John-David T. Smith on Wed, 05 Jul 2000 07:00:00 GMT
View Forum Message <> Reply to Message

Craig Markwardt wrote:

```
>
> "J.D. Smith" <jdsmith@astro.cornell.edu> writes:
>>
>> void=routine_names('a',STORE=-1,val)
>>
>> Lions and tigers and bears, oh my!
>
> Pay no attention to the man behind the curtain!  I still say this
> doesn't work for most cases, at least under Unix.  The following is a
> simple transcript to test thes, which shows how ROUTINE_NAMES crashes
> when A is undefined, but works when A is defined.  In every case that
> I tried (as noted below), the variable A needs to be defined at the
> calling level *first* before ROUTINE_NAMES(STORE=) will work.
>
> It could be that this works under v5.3, or on Windows.  I don't know
> since we don't have those versions around here.
>
> Craig
>
> IDL> .run
> pro testset, name
> void = routine_names(name,store=-1,findgen(10))
> end
> % Compiled module: TESTSET.
>
> IDL> testset, 'a'   ;; A is undefined here, and it doesn't work
> % ROUTINE_NAMES: Variable is undefined: a.
> % Execution halted at:  TESTSET             2 /dev/tty
> %                       $MAIN$
> IDL> retall
>  IDL> a = 1
>  IDL> testset, 'a'   ;; Now A is defined and it works.
>  IDL> help, a
>  A               FLOAT     = Array[10]
>
>  IDL> print, !version
>  { x86 linux unix 5.2.1 Jun  4 1999}
>  { alpha OSF unix 5.2 Oct 30 1998}
>  { x86 linux unix 5.1 Apr 13 1998}
>  { alpha OSF unix 5.0 Apr 28 1997}
>  { alpha OSF unix 4.0.1}
```

```
IDL> testset,'a'
IDL> print,a
      0.00000      1.00000      2.00000      3.00000      4.00000      5.00000
      6.00000      7.00000      8.00000      9.00000
IDL> print,!VERSION
{ x86 linux unix 5.3 Nov 11 1999}
```

You just need the new magic wand, version 5.3.

JD

--
 J.D. Smith                              /*\    WORK: (607) 255-6263
 Cornell University Dept. of Astronomy  */      (607) 255-5842
 304 Space Sciences Bldg.               /*\     FAX: (607) 255-5875
 Ithaca, NY 14853                       */

## Subject: Re: Multiple values from a function?
## Posted by Simon de Vet on Wed, 05 Jul 2000 07:00:00 GMT

View Forum Message <> Reply to Message

Martin Schultz wrote:

>  Simon de Vet wrote:
>>
>> [...]
>>  I have managed to solve my problem, but in a way unlike anything mentioned
>>  here so far.
>>
>  Not true: this is exactly what I proposed.

Wah!

I had read your post, but at the time I didn't realise that it was using
structures. I later figured this out on my own, but didn't bother to re-read

the messages.

It was "keyword parameters" that threw me off...

:(

Simon

Subject: Re: Multiple values from a function?
Posted by Craig Markwardt on Wed, 05 Jul 2000 07:00:00 GMT
View Forum Message <> Reply to Message

"J.D. Smith" <jdsmith@astro.cornell.edu> writes:
>
>  void=routine_names('a',STORE=-1,val)
>
>  Lions and tigers and bears, oh my!

Pay no attention to the man behind the curtain!  I still say this
doesn't work for most cases, at least under Unix.  The following is a
simple transcript to test thes, which shows how ROUTINE_NAMES crashes

I tried (as noted below), the variable A needs to be defined at the
calling level *first* before ROUTINE_NAMES(STORE=) will work.

It could be that this works under v5.3, or on Windows.  I don't know
since we don't have those versions around here.

Craig

```
IDL> .run
pro testset, name
void = routine_names(name,store=-1,findgen(10))
end
% Compiled module: TESTSET.

IDL> testset, 'a'   ;; A is undefined here, and it doesn't work
% ROUTINE_NAMES: Variable is undefined: a.
% Execution halted at:  TESTSET             2 /dev/tty
%                       $MAIN$
IDL> retall
IDL> a = 1
IDL> testset, 'a'   ;; Now A is defined and it works.
IDL> help, a
A               FLOAT     = Array[10]
```

```
IDL> print, !version
{ x86 linux unix 5.2.1 Jun  4 1999}
{ alpha OSF unix 5.2 Oct 30 1998}
{ x86 linux unix 5.1 Apr 13 1998}
{ alpha OSF unix 5.0 Apr 28 1997}
{ alpha OSF unix 4.0.1}
```

--
 ------------------------------------------------------------ --------------
Craig B. Markwardt, Ph.D.         EMAIL:    craigmnet@cow.physics.wisc.edu
Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response
 ------------------------------------------------------------ --------------

### Subject: Re: Multiple values from a function?
### Posted by John-David T. Smith on Wed, 05 Jul 2000 07:00:00 GMT

View Forum Message <> Reply to Message

Martin Schultz wrote:
>
> Simon de Vet wrote:
>>
>> [...]
>> I have managed to solve my problem, but in a way unlike anything mentioned
>> here so far.
>>
> Not true: this is exactly what I proposed. I would be amazed if you
> would find a way that is entirely different (and still usable) from the
> recommendations that were provided to you (not counting some vague
> possibility of using common blocks, of course).
>

But we lowlanders should not forget the sorcerer's apprentice who scrawled this epigram on the stone wall of his tower prison just before dissappearing forever:

```
void=routine_names('a',STORE=-1,val)
```

Lions and tigers and bears, oh my!

JD

### Subject: Re: Multiple values from a function?
### Posted by Martin Schultz on Wed, 05 Jul 2000 07:00:00 GMT

View Forum Message <> Reply to Message

Simon de Vet wrote:
>
>
> I have managed to solve my problem, but in a way unlike anything mentioned
> here so far.
>

Not true: this is exactly what I proposed. I would be amazed if you
would find a way that is entirely different (and still usable) from the
recommendations that were provided to you (not counting some vague
possibility of using common blocks, of course).

> [...][/color]

Martin

```
--
 [[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[ [[[[[[[
[[ Dr. Martin Schultz   Max-Planck-Institut fuer Meteorologie    [[
[[                     Bundesstr. 55, 20146 Hamburg            [[
[[                     phone: +49 40 41173-308                 [[
[[                     fax:   +49 40 41173-298                 [[
[[ martin.schultz@dkrz.de                                      [[
 [[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[ [[[[[[[
```

Subject: Re: Multiple values from a function?
Posted by Simon de Vet on Wed, 05 Jul 2000 07:00:00 GMT
View Forum Message <> Reply to Message

Simon de Vet wrote:

> It also produces a number of other arrays which I'd like to have access
> to. Unfortunately, I have discovered that a function can only return
> one value.

I have managed to solve my problem, but in a way unlike anything mentioned
here so far.

In the end, I managed to figure out how structures work. I put all the
arrays and variables into one structure at the end of my function, and
passed this to the main program. Here, I was able to take it apart again,
and get all the original arrays and variables back.

I don't know if this is a better or worse method than those mentioned, but
it works so I'll keep using it for now.

Simon