Subject: Re: Binary (0s and 1s) output
Posted by David Fanning on Tue, 07 Dec 2004 16:20:08 GMT
View Forum Message <> Reply to Message

josegomez@gmx.net writes:

> I'm very new to IDL, and I'm trying to convert 5 integers into a series
> of 40 logical (0/1) values. I have done this in fortran in the past
> using an internal write, and then converting each element in the output
> string into a 0 or 1. Given my limited knowledge of IDL, I tried to use
> print with a format specifier, as follows:
> IDL> print,format='(2B8.8)',0,255
> % Unexpected text in format.
> "(2B8.8)"
> ^
> % Execution halted at: $MAIN$
>
> According to my notes, the B format is there and can be used (this is
> IDL 6.0).
>
> The other problem is how to convert the (potentially!) resulting test
> string into an array of integers.

Well, this is new to me, but it seems to work here
on IDL 6.1.1 in Windows:

IDL> print,format='(2B8.8)',0,255
0000000011111111

Cheers,

David
--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/

Subject: Re: Binary (0s and 1s) output
Posted by Benjamin Hornberger on Tue, 07 Dec 2004 16:20:10 GMT
View Forum Message <> Reply to Message

josegomez@gmx.net wrote:
> Hi!
> I'm very new to IDL, and I'm trying to convert 5 integers into a series
> of 40 logical (0/1) values. I have done this in fortran in the past
> using an internal write, and then converting each element in the output
> string into a 0 or 1. Given my limited knowledge of IDL, I tried to use

> print with a format specifier, as follows:
> IDL> print,format='(2B8.8)',0,255
> % Unexpected text in format.
> "(2B8.8)"
> ^
> % Execution halted at: $MAIN$
>
> According to my notes, the B format is there and can be used (this is
> IDL 6.0).
>
> The other problem is how to convert the (potentially!) resulting test
> string into an array of integers.
>
> Many thanks
>

I think the "B" format code was introduces in IDL 6.1.

Benjamin

Subject: Re: Binary (0s and 1s) output
Posted by David Fanning on Tue, 07 Dec 2004 16:21:46 GMT
View Forum Message <> Reply to Message

David Fanning writes:

> Well, this is new to me, but it seems to work here
> on IDL 6.1.1 in Windows:
>
> IDL> print,format='(2B8.8)',0,255
> 0000000011111111

Oh, right. It does cause an error in IDL 6.0. Probably
wasn't ready for prime time then. :-(

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/

Subject: Re: Binary (0s and 1s) output

Posted by josegomez on Tue, 07 Dec 2004 17:15:58 GMT
View Forum Message <> Reply to Message

David Fanning wrote:
> David Fanning writes:
>
>> Well, this is new to me, but it seems to work here
>> on IDL 6.1.1 in Windows:
>>
>> IDL> print,format='(2B8.8)',0,255
>> 0000000011111111
>
> Oh, right. It does cause an error in IDL 6.0. Probably
> wasn't ready for prime time then. :-(

Hmmmm... That's what I thought. I guess I'll need to do this in 6.1,
and I can just then use the READS command to convert to an integer
array.

Many thanks for your help, all who replied!
Jose

Subject: Re: Binary (0s and 1s) output
Posted by marc schellens[1] on Wed, 08 Dec 2004 05:57:24 GMT
View Forum Message <> Reply to Message

josegomez@gmx.net wrote:
> Hi!
> I'm very new to IDL, and I'm trying to convert 5 integers into a series
> of 40 logical (0/1) values. I have done this in fortran in the past
> using an internal write, and then converting each element in the output
> string into a 0 or 1. Given my limited knowledge of IDL, I tried to use
> print with a format specifier, as follows:
> IDL> print,format='(2B8.8)',0,255
> % Unexpected text in format.
> "(2B8.8)"
> ^
> % Execution halted at: $MAIN$
>
> According to my notes, the B format is there and can be used (this is
> IDL 6.0).
>
> The other problem is how to convert the (potentially!) resulting test
> string into an array of integers.

I think this printing/reading is quite ugly.

Better do something like:

bytIn = [your byte value array]

nIn = n_elements(bytIn)

boolOut = bytarr(8, nIn)

for bit=0,7 do begin

boolOut[bit, *] = (bytIn and 2^bit) gt 0

endfor

boolOut[bitnumber, inputIntegerNumber] contains now your result

HDH,
marc
