

---

Subject: Re: A Simple IDL Manifesto

Posted by [David Fanning](#) on Mon, 17 Nov 2008 18:01:34 GMT

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

---

David Fanning writes:

- > Folks,
- >
- > On my way to work this morning, after I my German
- > lesson gelearnt, I started thinking about simple
- > things ITTVIS could do to improve the IDL experience
- > for old and new users of IDL alike.

Gosh, lots of typos in that post. I didn't realize that when you start mixing word order up in your head that it would spill over into the spelling parts of your brain, too. :-(

Cheers,

David

P.S. Let's just say I hate to think what would happen if I started studying Arabic and had to read right to left. :-(

--

David Fanning, Ph.D.

Coyote's Guide to IDL Programming ([www.dfanning.com](http://www.dfanning.com))

Sepore ma de ni thui. ("Perhaps thou speakest truth.")

---

---

Subject: Re: A Simple IDL Manifesto

Posted by [Kenneth P. Bowman](#) on Mon, 17 Nov 2008 20:40:24 GMT

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

---

In article <MPG.238b5cb02f78dc04989768@news.giganews.com>, David Fanning <[news@dfanning.com](mailto:news@dfanning.com)> wrote:

- > 7. Make the NAN keyword default to 1 everywhere. Why not!?

Maybe my earlier post has made it around yet.

I think this is a bad idea. One of the fundamentally great insights in the IEEE standard is that NaNs and Infs propagate through calculations.

I don't want the default IDL behavior to be "hide the existence

of NaNs". This is a good way to create errors that are very difficult to track down, or worse yet, are never found at all.

Cheers, Ken

---

---

Subject: Re: A Simple IDL Manifesto

Posted by [lbnc](#) on Tue, 18 Nov 2008 06:18:12 GMT

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

---

On 17 Nov, 17:50, David Fanning <n...@dfanning.com> wrote:

> Folks,

>

> On my way to work this morning, after I my German  
> lesson gelearnt, I started thinking about simple  
> things ITTVIS could do to improve the IDL experience  
> for old and new users of IDL alike. My criteria for  
> my list is that it had to be something a junior ITTVIS  
> software engineer could accomplish in a lazy afternoon,  
> while knocking back a couple of cold Brewskis.

>

> Here is the list:

>

> 1. Make the COLOR keyword default to 1 for the PostScript  
> device. The last time I wanted two-color PostScript output  
> was in 1964, and if I want it now I can draw in black  
> and white colors.

>

> 2. Make the BITS\_PER\_PIXEL keyword for the PostScript  
> device default to 8. Setting it to 4 probably made  
> sense in the 18th century (I'm speaking metaphorically  
> in computer time, of course), but it makes absolutely no  
> sense today.

>

> 3. Get rid of the TRUE keyword on the TV command. If you  
> don't know how to tell if an image is a 2D or a true-color  
> image, I'm certain Liam can enlighten you.

>

> 4. Add a POSITION keyword to the TV command so you don't  
> have to spend an afternoon figuring out who to overlay an  
> image with some information.

>

> 5. Oh, the hell with it, ditch the TV command and incorporate  
> TVIMAGE or IMGDISP in it's place. (Personally, I would vote for  
> IMGDISP.) That way you could have axes and everything else, too.

>

> 6. Make square-bracket array subscripting the default. Craig  
> will get used to it, believe me.

>  
> 7. Make the NAN keyword default to 1 everywhere. Why not!?  
>  
> 8. I realize it is difficult to tell when you are staring at  
> a piece of paper just ejected from the printer, but make the  
> LANDscape PostScript output LANDscape and not SEAscape. You can  
> tell, all the time, when you view the PS output in GhostView.  
>  
> 9. Make UNXI computers default to TRUECOLOR visuals instead of  
> DIRECTCOLOR. Oh, wait... this is already done. Hurray! Scratch  
> this one from the list.  
>  
> Cheers,  
>  
> David  
> --  
> David Fanning, Ph.D.  
> Coyote's Guide to IDL Programming ([www.dfanning.com](http://www.dfanning.com))  
> Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Can I add:

10) Do *\*not\** throw an error when I want to plot just NaNs. Instead, print a warning and continue, maybe plotting only the axis. And David, don't suggest I should get children to understand that there is "always a hurdle to get over" - tried that, didn't help. Apparently didn't help for you either...

Cheers  
Lasse

---

Subject: Re: A Simple IDL Manifesto  
Posted by [David Fanning](#) on Tue, 18 Nov 2008 17:37:47 GMT  
[View Forum Message](#) <> [Reply to Message](#)

Kenneth P. Bowman writes:

> I think this is a bad idea. One of the fundamentally  
> great insights in the IEEE standard is that NaNs and Infs  
> propagate through calculations.

Alright, I'll concede this point. But since the engineer will have some free time, I thought of one more thing to add:

10. Add Brewer color tables to IDL. Once you start working with Brewer color tables, IDL's normal color tables seem so,

well, jarring. Like turning on the radio one morning to hear your favorite smooth jazz station only to find they have switched the format to hip-hop. It makes you a little queasy.

Cheers,

David

--

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

---

---

Subject: Re: A Simple IDL Manifesto  
Posted by [pgrigis](#) on Tue, 18 Nov 2008 19:08:42 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

David Fanning wrote:

> Kenneth P. Bowman writes:

>

>> I think this is a bad idea. One of the fundamentally  
>> great insights in the IEEE standard is that NaNs and Infs  
>> propagate through calculations.

>

> Alright, I'll concede this point. But since the engineer  
> will have some free time, I thought of one more thing  
> to add:

>

> 10. Add Brewer color tables to IDL. Once you start working  
> with Brewer color tables, IDL's normal color tables seem so,  
> well, jarring. Like turning on the radio one morning to  
> hear your favorite smooth jazz station only to find they  
> have switched the format to hip-hop. It makes you a little  
> queasy.

I am surprised you did not mention fonts in direct graphics...  
Has the issue been solved while I wasn't looking?

Ciao,  
Paolo

>

> Cheers,

>

> David

>

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

---

---

Subject: Re: A Simple IDL Manifesto  
Posted by [David Fanning](#) on Tue, 18 Nov 2008 19:13:18 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Paolo writes:

> I am surprised you did not mention fonts in direct graphics...  
> Has the issue been solved while I wasn't looking?

I've given up. Apparently, this is "impossible". :-(

Cheers,

David

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

---

---

Subject: Re: A Simple IDL Manifesto  
Posted by [R.Bauer](#) on Tue, 18 Nov 2008 20:41:32 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

David Fanning schrieb:

> Folks,  
>  
> On my way to work this morning, after I my German  
> lesson geleart, I started thinking about simple  
> things ITTVIS could do to improve the IDL experience  
> for old and new users of IDL alike. My criteria for  
> my list is that it had to be something a junior ITTVIS  
> software engineer could accomplish in a lazy afternoon,  
> while knocking back a couple of cold Brewskis.  
>  
>

12.) add utf-8 support for direct graphics and add some coding support.  
stop to implement it only for itools.

cheers  
Reimar

---

---

Subject: Re: A Simple IDL Manifesto  
Posted by [weitkamp](#) on Tue, 18 Nov 2008 20:56:37 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Reimar Bauer wrote:

- > 12.) add utf-8 support for direct graphics and add some coding support.
- > stop to implement it only for itools.

13.) Add the "dimension" positional parameter (as present in TOTAL) to MEAN, STDDEV and so forth.

---

---

Subject: Re: A Simple IDL Manifesto  
Posted by [R.Bauer](#) on Tue, 18 Nov 2008 21:34:41 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

14.) make it possible that we can have the last dimension 1.  
Everyone who dislikes it can use reform. I can't remember how often this misbehaviour has created funny results (bugs).

Reimar

---

---

Subject: Re: A Simple IDL Manifesto  
Posted by [Jeremy Bailin](#) on Wed, 19 Nov 2008 14:28:40 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

- > 13.) Add the "dimension" positional parameter (as present in TOTAL) to
- > MEAN, STDDEV and so forth.

Hear hear!

- > 14.) make it possible that we can have the last dimension 1.
- > Everyone who dislikes it can use reform. I can't remember how often this
- > misbehaviour has created funny results (bugs).

I don't know... about half the time, I love how it works now, and the other half the time (like when I was debugging last night) I swear ferociously at it. In any case, that almost certainly falls under the category of would-break-too-much-existing-code.

-Jeremy.

---

---

Subject: Re: A Simple IDL Manifesto  
Posted by [wlandsman](#) on Wed, 19 Nov 2008 17:24:59 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

On Nov 18, 3:56 pm, weik...@esrf.fr wrote:

> Reimar Bauer wrote:  
>> 12.) add utf-8 support for direct graphics and add some coding support.  
>> stop to implement it only for itools.  
>  
> 13.) Add the "dimension" positional parameter (as present in TOTAL) to  
> MEAN, STDDEV and so forth.

And I'll add the single most important feature most requested by  
astronomers....

14.) Make an easy way to draw the "sun symbol" (circle with a dot in  
the middle) in a postscript font.

---

---

Subject: Re: A Simple IDL Manifesto  
Posted by [R.Bauer](#) on Thu, 20 Nov 2008 09:11:19 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Jeremy Bailin schrieb:

>> 13.) Add the "dimension" positional parameter (as present in TOTAL) to  
>> MEAN, STDDEV and so forth.  
>  
> Hear hear!  
>  
>> 14.) make it possible that we can have the last dimension 1.  
>> Everyone who dislikes it can use reform. I can't remember how often this  
>> misbehaviour has created funny results (bugs).  
>  
> I don't know... about half the time, I love how it works now, and the  
> other half the time (like when I was debugging last night) I swear  
> ferociously at it. In any case, that almost certainly falls under the  
> category of would-break-too-much-existing-code.  
>  
> -Jeremy.

Hi Jermey

idl is an interpretive language. It is parsed line by line.

would-break-too-much-existing-code is a wrong statement. It needs only more effort.

If one changes the code rules behaviour he can also write a migration tool which converts your old program into a better new program.

But this wasn't also done by this company in the past, so we can assume it won't be in the future.

And it is not unusual to do so. e.g. if the moinmoin wikisoftware project changes the wiki text syntax in a newer version we provide a migration tool for the old wiki text syntax on pages to the new syntax.

cheers  
Reimar

---

---

Subject: Re: A Simple IDL Manifesto  
Posted by [Michael Galloy](#) on Thu, 20 Nov 2008 16:42:10 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

On Nov 20, 2:11 am, Reimar Bauer <R.Ba...@fz-juelich.de> wrote:

- > If one changes the code rules behaviour he can also write a migration
- > tool which converts your old program into a better new program.
- >
- > But this wasn't also done by this company in the past, so we can assume
- > it won't be in the future.
- >
- > And it is not unusual to do so. e.g. if the moinmoin wikisoftware
- > project changes the wiki text syntax in a newer version we provide a
- > migration tool for the old wiki text syntax on pages to the new syntax.

Yes, a similar change like this is happening for Python 3000 i.e. Python 3.0. They are providing a py2to3 tool that will convert Python 2.6 to Python 3.0 code. This tool will do most of the grunt work, but I believe some hand coding will still be necessary.

Backwards compatibility is a noteworthy goal, but the design of any language will eventually show its age. I think eventually you have to change things that turned out to be mistakes (hindsight is 20/20).

The things that I would change about the core language of IDL (not the library) that would break backward compatibility would be:

1. get rid of that extra "blankity, blank" comma when calling a procedure (the one right after the name of the procedure)

2. consistent handling of arrays with a last dimension of 1 (don't remove dimensions for me, thank you)

3. allow arrays of length 0

4. make "compile\_opt idl2, logical\_predicate" the default

A conversion tool could probably do 1 and the "idl2" part of 4 pretty easily. 2, 3, and the "logical\_predicate" of 4 would be a bit harder and probably require some overview by the developer.

Of course, this means that code written for this new "IDL 8" would not work in previous versions (the tool would only convert from old to new style). If the .sav file format didn't change, then at least "IDL 8" .sav files could be used in previous versions.

Another solution would more compile\_opt flags, but I'm not sure what should happen if a routine with the new compile\_opt flag had an array of length 0 and passed it to a routine without the new compile\_opt flag. And I'm getting tired of putting a compile\_opt statement in every routine I write.

Mike

--

[www.michaelgalloy.com](http://www.michaelgalloy.com)

Tech-X Corporation

Associate Research Scientist

---