

---

Subject: Re: Top 10 IDL Requests

Posted by [bradgom](#) on Tue, 25 Jul 2000 07:00:00 GMT

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

---

In article <397876AD.283E645D@kla-tencor.com>,

Eric Vella <eric.vella@kla-tencor.com> wrote:

- > Here's a more ambitious request, which I don't expect RSI to
- > consider, but what the heck: Rewrite IDL as a Perl module. The IDL
- > syntax is really annoying, with all sorts of hard to remember quirks and
- > strange behavior everywhere.

Well, someone who is locked in Perl-mode is bound to think this way. The converse is also true- a trained IDL user will find Perl to be extremely quirky and counterintuitive. Both languages represent distinctly different approaches, and are not necessarily equally well suited to all tasks.

- > This was not so bad when IDL was young and
- > had just a few simple commands, but in recent years RSI has expanded IDL
- > into a full-blown language. This is a classic example of reinventing
- > the wheel, badly. I have no interest in learning a complete new computer
- > language, so I will never be an advanced IDL user -- I will just use

When it comes to things like parsing strings, Perl is pretty good. When it comes to any kind of analysis involving arrays, IDL is superior -both from the standpoint of syntax simplicity, and computational efficiency. The Perl Data Language is an attempt to add IDL functionality to Perl, and while it is useable, it is a far more blatant case of 'reinventing the wheel, badly'.

Having learned C and IDL first, I am probably biased against Perl, but I suspect if you took an average student who had never programmed before, and asked them to write a quick routine to do some data visualization in PDL and IDL, they would have a far easier time with IDL.

If you don't want to learn any new syntax different from Perl, you will probably have a hard time finding any other language to use.

Brad Gom

Sent via Deja.com <http://www.deja.com/>  
Before you buy.

---