

---

Subject: Re: IDL vs Yorick?

Posted by [Craig Markwardt](#) on Fri, 11 Oct 2002 14:52:55 GMT

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

---

Ralf Flicker <ralf@astro.lu.se> writes:

- > At the risk of reiterating an old debate (if there was one), I
- > would like to hear people's opinions about the open source
- > interpreted language called Yorick.

Greetings Ralf--

I meant to answer this a long time ago, but it just scrolled too far.

I am not surprised that Yorick can outperform IDL in some areas. The author, David Munro, has put a lot of effort into scalability.

Also, Yorick has some wonderful variations on the IDL syntax. While for the most part Yorick is very close to IDL, there are some areas, like array indexing, where Yorick is clearly superior. There are all sorts of novel ways to index arrays. And things like my CMAPPLY or JD's median or variance kludges come naturally because you can apply functions directly to the dimensions of an array.

That being said, I think that Yorick is currently not a viable candidate for me, and I'll say why:

1. Close-to-non-existent user community. There's no mailing list or newsgroup that I know of.
2. Close-to-non-existent development. I can see that Munro is slowly developing a new version, 2.0, but this has happened over 4 or 5 years. This is not to mean any discredit to Munro! Quite the opposite. His is fantastic design, but unfortunately it's only him as far as I can see.
3. Poor debugging. I never figured out how to debug from the command line.

And the most important reasons:

4. Little or no third-party libraries. Of course this is self-defeating :-) There won't be third party libraries until people develop for it. On the other hand, the very basic things like curve fitting are missing, or are tack-ons.
5. My own huge sunk investment in IDL code. I have something like 100k lines of code written in IDL in my personal library, plus lots

of custom scripts etc. That would be 98% wasted if I switched to Yorick.

Reason number 5 is certainly the most frustrating, and yet like a crack cocaine addict, I come back for more.

Reasons 4 and 5 have led me to conclude that the only viable alternative to IDL must at the very least have some form of "compatibility mode" which runs 99.5% to 99.9% of existing code with no changes. And I'm serious about those percentages. With a 100k line library, I don't want to be making more than a hundred or so changes to be compatible with something else.

Craig

---