
Subject: IDL 7.0 Preview Review

Posted by [David Fanning](#) on Wed, 14 Nov 2007 05:47:11 GMT

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

Editor's Note: I apologize in advance. We had assigned our most reputable technology reporter, David Fanning, to this story, but he has come down with "food poisoning" and has taken to his bed. (At 7:30, no less!) Because we are on a tight publication deadline, we have asked our Entertainment Reporter, El Senor Coyote, to fill in for Mr. Fanning. Unfortunately, Mr. Coyote did not attend the entire presentation. (He says he found the three hour business portion of the meeting intolerable without any breaks.) What follows, then, is what Mr. Coyote has been able to cobble together from Mr. Fanning's hastily written notes and his own, imperfect memory. We apologize if events didn't transpire exactly as reported here.

Boulder, CO 13 Nov 2007 --

Excitement was in the air, and programming testosterone filled the room, as the lucky guests at the IDL 7.0 Preview and User's Group Meeting were ushered along the red carpet and into their seats shortly before 1:00. The question on everyone's mind was: What does the damn thing look like!? They were referring, of course, to the new Eclipse-based IDL development environment, now called the IDL Workbench.

We had to wait nearly 20 minutes to find out, while we listened to a long history of IDL. (It first got its start in the punch card era, I guess.) Luckily, David Stern was in the audience to confirm that "Yes, that's how it happened." But that was then, and this is now, and **finally** we have found the one true course, and we can expect nothing but great things in the future for IDL. (Or something to that effect. I had to take a bathroom break.)

Anyway, the new IDL Workbench sorta looks like the old IDLDE, but not quite, and its WAY cooler than anything else anyone could think of, and it works the same on every platform God ever made, which is a HUGE advantage to the IDL developers, as they only have to maintain one code base now. (Presumably freeing up some time to work on something else, so we will get more for our maintenance dollars, and so forth.)

One of the developers showed us a few of its tricks. (A still secret incantation will allow the user to hook up an RSS feed

of this very newsgroup into the Workbench environment. A feature some people may not see as much of an advance, come to think of it.)

Anyway, LOTS of new features that certainly got the presenters pretty darned excited. Of course, the cynical programmer sitting beside me pointed out that he had been doing everything they were showing up there for *at least* the last five years with EMACS and IDLWAVE mode. So I guess you could think of the IDL Workbench as EMACS-lite, if you like. Anyway, it does some nifty things.

Of course, if you are too old to change (it is a possibility for some of the gray hairs sitting in the audience), those of you working on UNIX platforms can still have access to the normal IDL command line. If you are a Windows user, well, you are used to being told what to do anyway. And, I'm SURE you are going to like it!

Everyone was careful to point out that there was a bit of a learning curve with the new Workbench, but that in a couple of days you would be so overjoyed with the new features that learning would actually be fun again. Ronn Kling has written a little book, entitled "Navigating the IDL Workbench" that I suppose is yours for the asking. The second chapter is ominously entitled "Don't Panic!!!", but it is short enough to read while you eat a burrito, so that's encouraging.

After the IDL Workbench demo we got an update on where IDL is headed in the next 3-4 years. The target audience "doesn't want to read a manual", "wants something that looks like his Java or C++ environment", "wants to focus on the task at hand", and "wants to get away from programming". And--I was thinking--is dumb as a stone, but they didn't say that.

Anyway, expect more features that "modernize how users interact with IDL". This would include more GUI-based tools, drag and drop functionality, and what is going to be called "Interactive Ad-Hoc Analysis", which is, well, different in some way than what we do now. "We want to change the way you interact with IDL!", they said. Hey, I'm all for that. The less thinking I have to do, the better, I say. They did offer some examples of how one might set graphics and other visualization properties from the IDL command line that looked to me a lot like what MatLab allows you to do now.

We did get some exciting news. The IDL GUI-Builder is history! Having that god-awful code languish in infamy is a step forward, I think. :-)

The bad news is, we are going to make another stab at a ToolBuilder sometime in the future. (There is hope we might get something useable on this, the fourth, or is it fifth?, iteration.) Anyway, Eclipse is

set up for this kind of thing, so we are going to have another go. This one will be neat. No event handling, no state structure, no coding. Everything will just work! (I asked how data you might want to do something with would get passed around in this "program", but those details haven't been worked out yet. In conversations later, I learned that people were actually thinking about this and there was talk about having data "show up" in the "right scope" somehow. At least it gave me some ideas for how I could make data "show up in the right scope" even now!)

After the Workshop presentation, we were treated to a demonstration of an IDL program written at LASP, using iTool functionality. Just about the entire thing had been coded up by an undergraduate programmer who "didn't know a darn thing about IDL when he started." The point of the presentation was, apparently, to demonstrate how easy it is to build tools using iTools functionality.

Of course, they couldn't quite work out how to build the iTool they wanted to build (and who here would cast the first stone?), so they opted for putting all their functionality in a single pane of an existing iTool. Still, it worked pretty good. Then we learned that "We use common blocks to pass data. It's probably not ideal, but it was one of the ways we could get around some iTool limitations." Whatever. I doubt a professional programmer could have done much better, and my hat is off to the kid.

Next up, Ronn Kling showed us some of the neat things he has coded up for his web page and in his annual Christmas cards. He is always doing *something* interesting. We didn't get to look at code, of course, and the "how?" was pretty sketchy, but sometimes just knowing something is possible is enough to give you encouragement.

Finally, an ITTVIS technical support engineer gave us a run-down on what's happening on the ITTVIS web page. We learned about some neat new things on the IDL Code Contrib page, including (and I have not downloaded this yet myself) an IDL program for viewing and reading GRIB files. (My cynical programming friend and I both raised our eyebrows, pretty much simultaneously, so that is a VERY good sign.)

I detected a new and refreshing openness on the part of IDL developers at this meeting to interact and engage customers in what they really wanted. There is an openness to working with 3rd party developers and consultants that I have never seen before. I certainly came away feeling that the ITTVIS people are smart, capable,

and ready to listen to user feedback.

Oddly, I don't seem to run into the customer they are apparently writing code for in my day to day IDL dealings, but he or she must be out there. They are sure spending a lot of time and money going in this direction.

I'm going to have a look at the new IDL Workshop. God knows I've tried to learn EMACS for the last 4-5 months and I **still** have to look up how to do a Search-and-Replace every time I want to do it. My fingers just don't want to learn that dance. The IDL Workshop may be just what the doctor ordered. At least I hope so.

The IDL 7.0 CD was signed off on just today. It will be available from the ITTVIS web page sometime near the end of November, and will be shipped to users with valid maintenance contracts about two weeks after it is released on the web page.

Faithfully rendered from Fanning's lousy notes,

Coyote

--

David Fanning, Ph.D.

Fanning Software Consulting, Inc.

Coyote's Guide to IDL Programming: <http://www.dfanning.com/>

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