Subject: IDL, GDL, copyright, EULAs and such
Posted by Y.T. on Wed, 06 Jul 2005 17:12:47 GMT
View Forum Message <> Reply to Message

So I've been playing around with GDL, simply because I like IDL. I like GDL's focus on the essentials - I wouldn't mind widget-tools, but I consider them a luxury, really. And if GDL never acquires the incomprehensible object junk of IDL, it'll be too soon.

However much of the power of IDL lies in the library, of course - and thus GDL's weakness lies in the lack of that library.

For now, I figure, there shouldn't be a problem with my re-using the routines from my valid, legal IDL installation -- it is just a massive pain when such an essential tool like "linfit" is unavailable. So I copy it from my /usr/local/rsi to a local GDL-directory. No big deal.

Or is it a big deal? The routine is copyrighted by RSI, no? Am I allowed to use it on a program like GDL that is quite blatantly intended to be a free replacement for IDL? Can I use my idl/lib routines as long as I have a functioning IDL installation? Do I have to stop using them as soon as my license expires? But the license is only for IDL, no? That's why the hasp-thingee only protects use of the binary, right? So the library should be considered "acquired" and still be allowed to be used with GDL even in the absence of a working IDL. No? Yes?

The longer I think about it the less sure I am that running GDL might not be in violation of the "reverse engineering" clause in the IDL license agreement or some such -- except that that would only apply to people who actually have a valid installation of IDL since only those would ever have agreed to that license.

And the IDL license actually never spells out what the "software" really is that is being licensed -- whether it is IDL itself or also the contents of idl/lib/*

Since I'm already pretty confused by the whole notion of "intellectual property" (and much more so on the 'net) I figure I'll throw this out as an open-ended question -- can I continue to use basic, trivial functions like "poly.pro" or "factorial.pro" with GDL after IDL has expired? How about more complex stuff? Why or why not? What's the status of the routines that were originally from NR?

I'm just baffled.

cordially

Y.T.

--
Remove YourClothes before you email me.

Subject: Re: IDL, GDL, copyright, EULAs and such
Posted by Y.T. on Thu, 14 Jul 2005 01:33:02 GMT
View Forum Message <> Reply to Message

Ken Mankoff wrote:
> On Wed, 13 Jul 2005, Y.T. wrote:
>>> But I think if you look around
>>> http://www.astro.washington.edu/deutsch/idl/htmlhelp/
>>>
>>> You'll find that almost all the IDL code has been duplicated by
>>> someone else somewhere else. You are probably allowed to use
>>> that.
>>
>> Maybe someone needs to start collecting these things and compile
>> them into a "free idl-alike .pro library". The value of something
>> like GDL would really increase drastically through this.
>
> Umm... did you follow the above link? Doesn't that constitute a
> "collection"?
>

Not in any useful sense -- it's just a bunch of pointers that are not at all sorted by "public-domain"-vs-"proprietary". I can follow these links and examine every single library that I find and keep only those that explicitely state that I'm allowed to use it without a valid RSI license, but that process would be a lot of work (and would be exactly what I meant with "collect all these").

And in the end, there's a lot of basic functionality that would probably barf because it relies on the presence of other routines that RSI supplied. Which are NOT public-domain.

Subject: Re: IDL, GDL, copyright, EULAs and such
Posted by Michael Wallace on Thu, 14 Jul 2005 15:56:55 GMT
View Forum Message <> Reply to Message

> But if an algorithm is that obvious you can use it of course.

> Just write it yourself and don't copy it.
> It will look different. Even if it is similar: I think nobody could
> convince a judge that you should pay anything because you used
> *similar* code, especially if a subroutine is almost trivial.
> But if you use the *same* code it might be different.

Well, there is one thing that could throw a wrench into the works. It all depends on what definition of "reverse engineering" RSI is using in their EULA. In general, reverse engineering is the simply figuring out how a particular software feature or function works when you do not have the source code available. Much of the time we think of reverse engineering as someone getting into the binary files, using decompilers and figuring things out that way. But, it can also apply to just figuring out how something works to the point that you can reproduce it in an independent work.

If an algorithm is purely mathematical and you can point to a source other than IDL where the algorithm was published, then the reverse engineering claim doesn't stand. However if you study IDL long enough and figure out how to duplicate some internal algorithm exactly, then they could claim you reverse engineered that particular feature, even though your source code and RSI's code are different.

-Mike

Subject: Re: IDL, GDL, copyright, EULAs and such
Posted by m_schellens on Thu, 14 Jul 2005 17:26:56 GMT
View Forum Message <> Reply to Message

I think you can find for every library routine published algorithms from some other sources. For scientific software it would not be good to use black(-box) magic.

marc

Subject: Re: IDL, GDL, copyright, EULAs and such
Posted by Michael Wallace on Thu, 14 Jul 2005 17:53:14 GMT
View Forum Message <> Reply to Message

m_schellens@hotmail.com wrote:
> I think you can find for every library routine published algorithms
> from some other sources. For scientific software it would not be good
> to use black(-box) magic.
>
> marc

>

Definitely not. I was just trying to make the distinction between published scientific algorithms and internal IDL algorithms (such as how plots get drawn).

-Mike

Subject: Re: IDL, GDL, copyright, EULAs and such
Posted by m_schellens on Thu, 14 Jul 2005 19:13:27 GMT
View Forum Message <> Reply to Message

You cited the part where I was talking about
routines written in GDL.

As how plots get drawn:
GDL uses the plplot library here (which IDL cannot use as it is under the GPL). Definitely as far away from reverse engineering (RE) as it gets.

For me that RE stuff is way overstretched.
If just doing something similar would be illegal RE: What about .net and mono? What about M$ Office and Openoffice? Unix and Linux? The GNU command line programs?...
AFAIK a programming language itself cannot be protected anyway. There are loads of examples of compilers/interpreters for several proprietary languages. Most popular probably Java. Anotherone popular is Octave.

Cheers,
marc

Subject: Re: IDL, GDL, copyright, EULAs and such
Posted by Michael Wallace on Thu, 14 Jul 2005 20:12:07 GMT
View Forum Message <> Reply to Message

> As how plots get drawn:
> GDL uses the plplot library here (which IDL cannot use as it is under
> the GPL). Definitely as far away from reverse engineering (RE) as it
> gets.

I was just using plotting as an example. I wasn't commenting on GDL or anything else that has a plot feature. I was attempting to say that if you created a program that mimicked IDL plotting *exactly*, down to the position of every last pixel, that might fall under the "reverse engineering" banner in the EULA.

> For me that RE stuff is way overstretched.

I agree. But it is amazing what some companies will try to do to you with the claim that you reverse engineered something of theirs when it's actually a completely new implementation. There have been way too many frivolous lawsuits on stuff like that.

> If just doing something similar would be illegal RE: What about .net
> and mono? What about M$ Office and Openoffice? Unix and Linux? The GNU
> command line programs?...

Doing something *similar* isn't reverse engineering. But doing something that looks *identical* to something else could be called reverse engineering. Here, I'm only speaking of large, complicated programs that a company would sell, or at least contain a restrictive EULA.

> AFAIK a programming language itself cannot be protected anyway. There
> are loads of examples of compilers/interpreters for several proprietary
> languages. Most popular probably Java. Anotherone popular is Octave.

A programming language itself can't be protected because a programming language is a specification rather than an actual product. You don't sell the specification; you sell the compiler or interpreter of the language.

All I was attempting to say with my original comment about "reverse engineering" is that some companies will attempt to stretch that definition to cover independent works which behave identically to their own product. While I do not agree with this definition, it is something that happens from time to time in the business world and we should be aware of it, especially if we are creating a product similar to one that already exists. If all your ducks are in a row, then it should be no problem whatsoever. I just wanted to add that word of caution to those asking the original questions regarding what makes certain software legal or illegal. I was only using GDL as an example; I wasn't trying to say or imply that GDL, either in part or on the whole, was a reverse engineered product.

-Mike