

# The Embedded Muse 32

Editor: Jack Ganssle ([jack@ganssle.com](mailto:jack@ganssle.com))

February 1, 1999

## The Trouble with Open Source

Last issue I went off about issues I have with the “open source” movement, generating a staggering volume of replies.

Several folks asked if the article was a troll, an attempt to get folks riled up. Well, *of course* it was a troll - there’s nothing more fun than hearing strong opinions from smart people... though that doesn’t mean I don’t think that some of the reasoning used for distributing source code is flawed.

Others accused me of being a Microsoft stooge, which is funny since my Windows 95 machine crashed three times in reading these emails!

Of the respondents, about a third agreed with me; two thirds were firmly at odds. There was no middle ground.

Many people made a number of very good points. To summarize the most common threads:

- Few developers trust the vendors. As one reader put it “I feel better trusting part-time hobbyists than the vendor to get it right.”
- No one forces you to modify the source, so there’s no downside to having it.
- Open source means you can port the code to another CPU or platform.
- The code from vendors is buggy. Developers need the code to fix it. One reader mentioned that 90% of coding time is debugging, especially with purchased code, so there’s a quality Vs source issue. Also, documentation errors often can’t be resolved without the source.

And so, since it’s only fair to give equal time to others, here are some excerpts from readers’ comments. All are reprinted with permission of the authors.

And thanks to everyone for writing. I wish there was room to share more of people’s thoughts.

Randy Brei writes:

*Copyright 2000 by The Ganssle Group. All Rights Reserved. You may distribute this for non-commercial purposes. Contact us at [info@ganssle.com](mailto:info@ganssle.com) for more information.*

The other day, one of my co-workers was singing the praises of Linux ... and one of (supposed) virtues was that you get all the source code. I'm thinking to myself, "that's the last thing I want to have: zillions of lines of source code". I've written an OS and I've studied a handful of other OSes. I don't want to work on another OS. I just want to be a user. Also, it is not a comforting feeling to me ... knowing that a thousands ... or millions ... of people have been messing about with the source code.

I particularly liked Bob Kodner's rephrasing of a Freak Brothers saying:  
Source and No Money will get you through the hard times better than Money and No Source.

Gary Bergstrom said:

I'm aware of the trap that many engineers fall into: NIH. and two corollaries: I can do it better & everyone else does it wrong. I try hard to not fall into these. What is important is getting the job done. If it's faster to work on my tools, so be it. If it's not, then I hope that my experience will let me know that before I waste time.

Donald Kerns and numerous others said:

The engineers get much better "kicks" releasing the "free" LINUX version (just because it's COOL) than the death marched commercial release.

John Ford spoke for a lot of readers:

I contend that if you rely on a vendor to fix bugs, you're doomed. I have NEVER had a vendor fix anything for me. I have found bugs in VxWorks, Solaris, and Microsoft products. I was never given a fix. I'm waiting right now (2 weeks!) for help on a bug in one of our purchased non-source products. If I had the source, I could fix it myself in a few hours.

From Scott Finneran:

I do however have a reason for wishing that I had the source to the cross-compiler that we use here. I am working on a project which has been going for about 10 years now (customers just keep asking for more functions... you know the story). Our cross-compiler (status: NO LONGER SUPPORTED) runs under an ancient version of an OS (status: NO LONGER SUPPORTED) on an old underpowered machine (status: NO LONGER SUPPORTED).

The answer is obvious... go back to the compiler vendor and get them to port it to the latest version of their OS. Their response to our requests is "that product is no longer

*Copyright 2000 by The Ganssle Group. All Rights Reserved. You may distribute this for non-commercial purposes. Contact us at [info@ganssle.com](mailto:info@ganssle.com) for more information.*

**The Ganssle Group, [www.ganssle.com](http://www.ganssle.com)**

supported". As they no longer sell the compiler, we requested the source so that we could port it ourselves. We were willing to sign NDAs and agree to only use it in house. The response: "that product is no longer supported". As we don't have the source to the compiler nor the budget to port our code to a new compiler (a massive effort due to compiler "quirks"), we are stuck. If however we had chosen for example EGCS - the Cygnus supported variant of GCC, we would be in far better shape. We could simply get any qualified contractor (for example Cygnus) to port the compiler to the new platform. Not only that, but we would be getting EXACTLY THE SAME COMPILER on our new blindingly fast platform. This also avoids the other problem of vendors coming back with "we have a new version which runs on your new platform..... their are just a couple of MINOR differences".

I guess what I am getting at is, no I don't want to tinker with the guts of my compiler but having the source is an insurance policy against my product outliving the tools that I use to build it.

Andrew Mayo wrote:

As someone who has been watching the recent emergence of the Open Source movement, I think you have been a little harsh on the idea of access to the source code.

While some misguided souls might indeed consider that they can hack into their compiler etc. and improve it, most of us don't want the source code for that reason.

Instead (and if you've ever tried to get support through Microsoft, for example, you'll see the point), the availability of Open Source means that support organisations will be able to properly resolve issues in a much more effective way. Suppose you have a bug in your compiler. You ring the support organisation (you are paying either per incident or via a contract, so they get to make some money, of course). The experienced techie at the other end can examine the source code to determine the problem and then either patch it, have you patch it, and/or notify the author(s) so that the next release will have the fix.

Whereas with Microsoft, the source code is locked up in Redmond. The support people in Singapore don't have access to it. So most problems take a lot longer to resolve. Would you rather have your bug fixed today or next year?

Open Source will clearly result in higher-quality software and quicker release cycles. In addition, access to the source will allow budding programmers to examine work done by professionals (hopefully). I submit that if you look, say, at the source for PGP or Perl, you are looking at quality software developed by experts. You will do well to copy their styles and techniques. Also, here are a million wheels already invented for you so you can just take the routines and re-use them. Want a really good random number generator? A hashing algorithm? A set of B-Tree routines?. Here they are.

*Copyright 2000 by The Ganssle Group. All Rights Reserved. You may distribute this for non-commercial purposes. Contact us at [info@ganssle.com](mailto:info@ganssle.com) for more information.*

Oh, and perhaps you'd like something running on another platform. Microsoft might not choose to do that for marketing reasons. For example, Internet Explorer 4 has been ported to Solaris but will we ever see it on Linux?. Not likely, I'd say. But Netscape's browser is publicly available in source form so of course it's already been ported.

Open Source is already a concept most embedded systems developers are familiar with, anyway. For example, take the PIC chip. Look at all the stuff Microchip provide on their web site in source form - clearly, this encourages developers to get started quickly. The Open Source movement simply takes this concept to an extreme.

Finally you have to look at the cost. With Linux, you could set up an engineering workstation a lot cheaper, spending the money on better design tools etc. Isn't this a good thing, or do you want Mr. Gates to earn even more than he does?

## **Thought for the Week**

See <http://www.duh-2000.com> for the latest on Y2K.

## **About The Embedded Muse**

The Embedded Muse is an occasional newsletter sent via email by Jack Ganssle. Send complaints, comments, and contributions to him at [jack@ganssle.com](mailto:jack@ganssle.com).

To subscribe, send a message to [majordomo@ganssle.com](mailto:majordomo@ganssle.com), with the words "subscribe embedded *your-email-address*" in the body. To unsubscribe, change the message to "unsubscribe embedded *your-email-address*".

The Embedded Muse is supported by The Ganssle Group, whose mission is to help embedded folks get better products to market faster. We offer seminars at your site offering hard-hitting ideas - and action - you can take now to ***improve firmware quality and decrease development time***. Contact us at [info@ganssle.com](mailto:info@ganssle.com) for more information.

*Copyright 2000 by The Ganssle Group. All Rights Reserved. You may distribute this for non-commercial purposes. Contact us at [info@ganssle.com](mailto:info@ganssle.com) for more information.*

**The Ganssle Group, [www.ganssle.com](http://www.ganssle.com)**