NEWS · JUNE 20, 2026

A Recommendation Is Not a Verdict

By Dr. Holger Flick



Let me be blunt: when I share a Delphi component on this blog, I am not handing down a judgment from the mountain. I am not declaring everything else obsolete. I am not saying I have never heard of your favorite alternative. I am a developer with thirty years in the Delphi world, I find something interesting or useful, and I tell you about it. That is the entire transaction.

So why does it so often feel like I kicked someone's dog?

What a Personal Recommendation Actually Is

When a colleague at a conference leans over and says "Hey, have you tried this component? It solved my problem nicely" — you do not look at them and say "Well, have you considered that ComponentX also does this and possibly better?" You say "Oh, interesting, I'll have a look." And then you either look or you don't.

That's all this blog is. A colleague leaning over. Someone with a bit of context who found something worth flagging. Nothing more, nothing less.

A personal recommendation carries exactly this weight:

- **I tried it, liked it, thought of you.** That's it.

- It is not a benchmark report.
- It is not an exhaustive market survey.
- It is not an implicit criticism of anything I didn't mention.

The Silence About B Says Nothing About B

Here is the part some readers seem to struggle with: if I write about Component A, my not mentioning Component B means absolutely nothing about Component B. It means I wrote about A. Full stop.

I have been in this community since before some of you started programming. I am aware that alternatives exist. I am aware that you have your favorites. I am genuinely glad they work for you. That does not mean every post needs a footnote listing every competing product to prove I did my homework.

If I wrote about every option every time, I would be writing documentation, not a blog. Nobody would read it, including you.

The Comments Section Is Not a Billboard

This is the part that genuinely puzzles me. Someone reads a post where I recommend a free, open-source component, and they use the comments to explain why their preferred commercial product is superior. This is not a contribution to the conversation. It is advertising disguised as feedback.

If you have a genuinely useful addition — "I've used this too, and here is a gotcha to watch out for" or "there's a related library that handles this edge case differently" — that is wonderful, post it. That is community.

Using the comments to make my suggestion look bad so your suggestion looks good is not community. We are not in kindergarten. We do not need to tear one thing down to lift another up.

The Value Is in the Trigger, Not the Verdict

The most useful thing a blog post can do is make you think: *"Hm, I never considered that approach."* It is meant to be a spark. What you do with the spark is entirely up to you. You might try the thing I mentioned and love it. You might try it and prefer something else. You might just file it away as "good to know this exists." All of those outcomes are valid.

What is not a valid outcome: feeling attacked because I shared something that wasn't your preference.

A Polite Request

If you find something on this blog useful — great, I am glad. If you disagree with a recommendation — also fine, feel free to share your experience. If you want to suggest an alternative in good faith — welcome.

But if your goal is to show up and demonstrate that I should have written about something different, or that my suggestion is naive, or that a real expert would have known about the thing you prefer: please reconsider. Save us both the time.

I will keep writing about what I find interesting. After thirty years I have earned the right to have opinions. And you have earned the right to take them or leave them.

That is how this works. That is how it has always worked.

INDEPENDENT · AD-FREE · READER-SUPPORTED

Free to read. Not free to make.

Every tutorial, article, and line of example code on Holger's Code is published without a paywall and without ads. This page is about what keeps it that way.

Professional work, published free

The content here is held to the same standard as professional client work: every tutorial is built as a real project, tested against current versions, documented step by step, and revisited when the frameworks move on. A single in-depth tutorial routinely takes days — and that's before the bills behind it: servers, storage, build infrastructure, licenses, and the hardware everything is verified on.

If an article or tutorial here has saved you an afternoon of debugging, taught you a technique, or helped you ship, it created real value for you. Supporting it isn't charity — it's paying for professional work you found useful, at whatever price you choose. And if you can't, keep reading anyway. That's what the free part means.

Ways to support

KO-FI

Buy me a coffee

A one-time tip — no account, no subscription, any amount.

BOOKS & COURSES

Buy the paid work

The paid tier of everything published free here — and you get something lasting in return.

SPREAD THE WORD

Share an article

Send a tutorial to a colleague, cite an article, subscribe to the RSS feed.

Nothing here goes behind a paywall, whether you chip in or not. But if something on this site earned its keep today, this is where to say so.

Support on Ko-fi —