NEWS · SEPTEMBER 30, 2025

Why You Need to Switch to Delphi 13 and its 64-bit IDE Yesterday

By Dr. Holger Flick

For most Delphi developers, the switch to 64-bit isn't obvious at first glance. After all, for **decades since Delphi 2**, we've been working quite happily in a **32bit IDE**, building applications, connecting to databases, and delivering production software.

With the introduction of 64 bit compilers for Windows and other platforms, the natural assumption was:

> *"As long as I can compile 64bit executables from the 32bit IDE, I'm fine— right?"*

Not quite. Let's talk about why sticking to the 32 bit IDE is holding you back, and why the time to switch is**right now**.

The Old World: 32bit IDE, 64bit Output

Today, Delphi allows you to build executables for multiple platforms, including 64 bit Windows. Technically, you could stay in the 32 bit IDE and compile flawless 64 bit executables. That's already been the case for years.

So why bother changing the IDE itself?

Because of one thing in particular: **database drivers.**

The Database Problem: Why the IDE Matters

Most Delphi developers working with databases rely on **FireDAC**. FireDAC integrates beautifully with the IDE, letting you connect to live data sources right at design time. This is crucial for productivity—you can design forms, test queries, and preview results without having to run your app every few minutes.

Here's the catch:

- A **32bit IDE can only talk to 32bit database drivers.**
- Many database vendors have already **stopped shipping 32bit drivers.**

That means if you're still working in the old 32 bit IDE:

- You **can't connect** to 64 bit-only databases inside the IDE.
- You're stuck waiting until runtime to test queries and data access.
- Your workflow becomes clunky, slow, and far less efficient.

In other words, Delphi might still compile your app, but you as the developer lose one of Delphi's greatest strengths: **live data at design time.**

Real-World Impact

Popular databases like **MySQL** and **PostgreSQL** have already moved to offering only 64 bit drivers. This trend is only accelerating.

If your daily work relies on design time database connectivity—and if you use FireDAC, it almost certainly does—then staying on the 32 bit IDE locks you out of modern database development.

Why You Should Switch Yesterday

Switching to **Delphi 13 and its Delphi 64bit IDE** ensures:

- Full FireDAC functionality at design time
- Compatibility with modern 64 bit database drivers
- Faster, more streamlined development workflows
- Future-proofing your projects as 32 bit support fades away

Yes, you can still technically build 64 bit apps with the 32 bit IDE. But unless you're okay with doing all your database work blind until runtime, it's no longer a practical choice.

Summary

The bottom line is simple:

- If you use **FireDAC** and connect to databases, the 64 bit IDE is no longer optional—it's essential.
- Many of the most popular databases (MySQL, PostgreSQL, and others) only offer 64 bit drivers now.
- The longer you postpone the switch, the more painful your development experience will become.

So don't wait. Switch to Delphi 13 with its 64bit IDE and compiler today.

Need Help?

If you're facing challenges with the transition, database integration, or optimizing workflows in Delphi, I offer **consulting and training services** to help teams modernize quickly and confidently.

Get in touch, and let's make your Delphi development future ready.

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 —