CodeGear: still on the wrong track? (http://www.sandon.it/node/11)

Posted on: Sat, 02/24/2007 - 23:10 **By:** Administrator

Recently, CodeGear (http://www.codegear.com) released its first new Windows products, Delphi 2007 for Win32 (http://www.codegear.com/Products/Delphi/Delphi2007forWin32/tabid/236/Default.aspx) and Delphi for PHP (http://www.codegear.com/Products/Delphi/DelphiforPHP/tabid/237/Default.aspx), after more than one year without anything new. Delphi for PHP is based upon Qadram Qstudio, and, despite it's name, it's not a new PHP personality within the Delphi/BDS IDE (Galileo). It's a separate product with its own IDE, although it was designed after the Delphi IDE. I wonder that someone thought it could be something to code PHP applications in Delphi (language), a real nonsense. Well, actually there are people thinking AJAX (http://en.wikipedia.org/wiki/Ajax_%28programming%29) is a language...

But let's focus on Delphi for Win32, the real "workhorse" (and cash cow...) in the products lineup. This release is Win32 only - it looks the whole Studio will be delivered later - but at a price of $1,299 (upgrade, Enterprise edition), it looks expensive, for a comparison the whole BDS 2006 upgrade price is $1,499.

What's new? Reading the press release, **Vista support**, **dbExpress 4**, a new (another!) framework to create Ajax web applications, MSBuild support, and lots of (unspecified) improvements. Very little, for a 1,299 upgrade.

From the information released, Vista support should mean glass frames and Vista's new dialogs - nothing more. From a screenshot, the hidden application form issues should be resolved - a long standing issue. Also, five years after the release of Windows XP, the IDE is themed (something not on my priority list) and hope applications will be too - wholly, remember the unthemed grid, for example?

dbExpress 4 is a complete rewrite (http://blogs.codegear.com/steveshaughnessy/archive/2007/02/16/31865.aspx) of the current dbExpress technology. Since the old Borland decided to deprecate the BDE and its SQL Links, they were not able to come up with a new, decent database access technology. When dbExpress appeared in Delphi 6, it was very buggy (it was a beginning of a trend, unluckily, as we discovered later). Development was very slow - for example Oracle TIMESTAMP datatype was not supported yet in BDS 2006, and many developer switched to ADO, or native library access components like DOA or ODAC. One of the crown jewels of Delphi, database access, didn't shine much anymore.

The new dbExpress should be Delphi code, and use the same code both in Win32 and .NET. I wonder if they could achive the same performance level writing both managed and native code. Database access frameworks are memory intensive, and memory management is one area where Win32 and .NET really diverge. Although the two compilers could hide it well enough, advanced techniques may not be used due to portability issues.

Moreover, the Delphi Win32 compiler didn't see real improvements in the past years. It still generates code based on 386 instruction set only, and does not take advantage of newer processors instruction sets - there is no switch to change this behaviour. Only thanks to an effort born outside Borland to overcome Delphi weakness in this area, the FastCode project (http://fastcodeproject.org/), CodeGear was able to integrate a far better memory manager in Delphi, and is integrating slowly a more performant RTL. FastCode could contribute to deliver better drivers, but as long as CodeGear does not update the compiler, the generated code is getting older and older - and that's not an issue for dbExpress drivers only.

MSBuild (http://msdn2.microsoft.com/en-us/library/wea2sca5.aspx) is a build techonlogy introduced by Microsoft, a kind of ANT (http://ant.apache.org/) for Windows. After twelve years - since the old days of TurboPascal 7.0 - we will be able again to save different build configuration. Again, CodeGear rely heavily on a Microsoft technology to improve its own development tool - if they did not want to write their own build tool, why didn't they integrate ANT - or the Delphi-written WANT - years before?

Overall, this looks to be a strange release. Some improvements, at an almost full upgrade price. Those getting it under their Software Assurance contract could find it useful, but an upgrade is hardly recommended unless someone really needs those features.

But the real issue is another. CodeGear did not show to be on the right track yet. The new care for the Win32 world is interesting, but right now it's the wrong care. It looks a kind of rearguard battle, a COBOL-like legacy care, aimed at those small shops who didn't switch to .NET or something else yet. Don't get me wrong. I believe native development is not going away, and there is room for a native RAD development tool. But it won't be filled by small shops with ageing programmers who don't want to learn .NET, Java, PHP or Ruby - they will run out soon - not because native application are wrong, just because their market will dictate so.
IMHO that space will be occupied by those needing an advanced development tool to write advanced applications with specific needs, that can't be addressed by P-code, GCed or dynamic languages. While till few years ago Delphi could have been a viable alternative to C++ for some kind of applications, it's now lagging behind too much. The lack of compiler improvements, 64 bit support, an update RTL, a better multitier framework, and so on, are now real issues for those who are beyond database frontends and intranet websites. We have outgrown Delphi, and it didn't keep our pace.

Ignoring the needs of this community means to focus on a shrinking market, and to look *old*. One of the big mistakes Borland made was to position Delphi at the same level of Visual Basic. It was much more poweful and innovative. Chasing the VB developer they lost the opportunity to position Delphi in an area where it had no real competitors. For simpler applications, the simpler-to-learn, MS-backed, and less expensive (developers included) VB was hard to defeat. Now MS raised the bar, and although not the "ultimate solution" MS tries to sell, .NET is a much more serious contender, looks *new*, is running fast - and don't forget - is cheaper.

CodeGear has still an opportunity to play well in the native application arena, but it needs to run fast too. Many of the issues above have promised solutions. Promised. Unluckily, I am not able to sell promises to my customers. I need to start to work today to build tomorrow's applications, not yesterday's.

- Log in (http://www.sandon.it/user/login?destination=/comment/reply/node/11/comment_node_blog%23comment-form) to post comments

Source URL: *http://www.sandon.it/node/11*