GExperts 1.0

Extend the Delphi environment for free

Reviewed by Primoz Gabrijelcic

Thanks to Borland, our favourite development tool is highly extensible with user-made plugins, or *experts* as they are usually called. But it is not trivial to write a new IDE expert, as most of the Delphi internals are almost undocumented and riddled with bugs that differ greatly depending on the version of Delphi used. That's why most of us don't even dream of writing an IDE expert. But still, we sure like to use them. Good IDE experts can make life with Delphi so much easier.

The greatest (in my view), and one of the oldest, of the Delphi IDE expert suites is GExperts. It was created by Gerald Nunn (in 1997, if I am not mistaken), who also developed it for almost two years. Gerald has moved to other projects, but he kindly donated the GExperts source code to the online community. The GExperts project

➤ *Figure 1: Most of the GExperts experts.*

is now lead by Erik B Berry, the internal framework is maintained by Stefan Hoffmeister, and individual experts are maintained by a bunch of contributors. You can get more information on the project at the main website at www.gexperts. org and also at the development website which is at sourceforge. net/projects/gexperts.

GExperts is free and is distributed as Open Source. You can always download the source, make any changes you want (and in the process learn more about writing experts) and contribute your work back to the community. Supported products are Delphi 3.02, 4.03 and 5 (Delphi 6 support is coming), plus C++Builder 4 and 5.

GExperts contains a lot, including: editor experts, toolbar buttons, component palette enhancements and code editor enhancements. Even more important, GExperts contains an entire framework for developing experts. It is relatively easy to add a new expert to the GExpert suite: just look into the other experts to see how they are implemented. Equally important is that the GExperts framework takes care of incompatibilities between Delphi versions and automagically fixes various problems with older Delphi releases.

Before we turn to the experts, we should take a quick look at the other GExperts additions, as they are also quite interesting.

Editor experts help the programmer with some common editing tasks: they will comment and uncomment the selected block of code, insert the date and time, locate a matching delimiter or move to it, and insert unit and procedure headers. Each editor expert can of course be assigned to any shortcut combination.

IDE enhancements enhance Delphi and C++Builder in different ways. There is multiline tab support for IDE dock hosts (only in Delphi 4 and higher), an auto-save option and the ability to save form files as text (Delphi 3 and 4 only). You can set custom fonts for the object inspector and component palette, enable the `Windows` menu (to list all the open windows), enable the menu item `Attach to Process` (in Delphi 4), and in Delphi 3 enable the menu items `Windows API` and `CPU View`.

Component palette enhancements can activate multiline tabs on the palette, turn tabs into (optionally flat) buttons, and add tab names to the palette popup menu.

Code editor enhancements improve the Delphi/C++Builder editor. Besides enabling multiline tabs (optionally displayed as buttons) and hot tracking, this part of GExperts creates a fully customisable editor toolbar in the editor window. This toolbar can contain native Delphi actions, GExperts experts, and drop-down lists containing all the forms and units for a project. Selecting a form or unit from the list activates the appropriate window.

Experts

And now, ladies and gentlemen, let's take a look at the core of the GExperts, its pride and honour: all 25 experts included in build 1.0. We won't be able to spend more than a few words on each, but, don't worry, on the GExperts website you can find a pretty good tour through the complete suite.

ASCII Chart shows ASCII values (integer or hexadecimal) for all the characters in the given code: simple and effective. *Backup Project* enables you to backup an entire project to a zip file: resource and include files are automatically included and you can also modify the list of files to be backed-up by hand.

Class Browser displays a browsable list of the classes in your project or in external directories, such as the VCL classes: reporting, filtering and quick jump to code are supported. *Clean Directories* enables you to quickly

delete several types of non-critical files that can accumulate during development (backup, object and temporary files, etc).

Clipboard History accumulates clipboard contents and allows you to view and use previously copied text. In Delphi 4 and above it is persistent across IDE sessions and dockable. *Code Librarian* is a simple code snippet organizer that stores its data in a Paradox database, which makes it a definite no-no for me: luckily, the next release will use `TClientDataSet` for data storage.

Code Proofreader can be most easily described as 'AutoCorrect for Delphi'. It can correct your typing in the code editor as-you-type, based on a list of custom dictionary words, a list of AutoCorrect entries, or context-sensitive information supplied by the Object Pascal compiler.

Component Grid shows a grid containing all the components on the form and presents their `Tag` and `HelpContext` properties for editing. *Components To Code* allows you to select one or more components on a form and copy the appropriate Object Pascal code to the clipboard that would create those same components at runtime. It can be a great help when creating composite components.

Expert Manager enables you to add, enable, disable, and remove DLL experts from the IDE without having to directly edit the registry. The change is not dynamic: you won't see the effect until you restart the Delphi.

Favorite Files is a very nice project organizer: you can create a tree structure of folders containing

➤ *Figure 2: IDE enhancements.*

items (projects and files) and other folders. *Grep Search* allows you to search through file or files for regular expressions and displays the results in the separate expert, *Grep Results*, that is dockable in Delphi 4 (and above) and allows quick navigation (double-click to jump to the matched line).

IDE Menu Shortcuts is another invaluable expert that enables you to configure almost any of the IDE menu shortcuts. *Message Dialog UI* will help you build those darned `MessageDialog` calls.
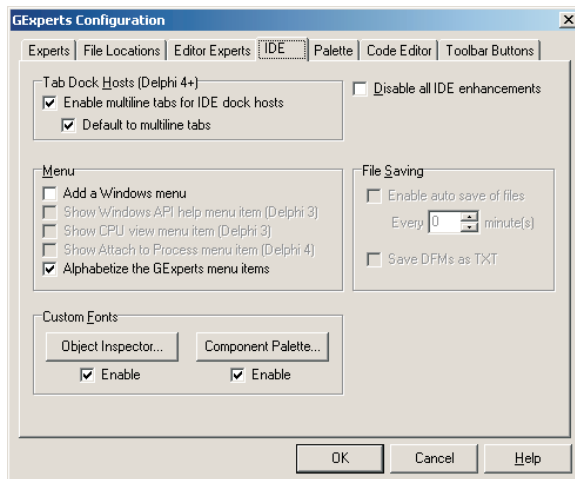
PE Information is strictly for experts: it allows you to examine the contents of the Portable Executable file (EXE, DLL, BPL, OCX, etc). *Perfect Layout* can be used to rearrange Delphi windows but has become obsolete with Borland's introduction of desktops in Delphi 5. *Priority Booster* modifies Delphi's priority while it is compiling the code. This should allow for faster builds and better Code Insight operation (Code Insight needs an internal compilation to operate).

Another of my favourite experts is *Procedure List*. It displays a browsable list of procedures in the active file. The list can be filtered by typing any substring of the procedure name and allows you to quickly jump to the selected procedure. It is great because it works in debug mode when Delphi's Ctrl+clicking is disabled.

Project Dependencies displays uses/is-used-by relations for all units in the open project.

Project Option Sets allows you to save project options into a named persistent storage and later

➤ *Figure 3: Project Dependencies expert.*

restore them. It works only in Delphi 4.02 and above, but actually Delphi 5.01 is required for full functionality because of some Delphi bugs. Potentially a useful tool, it is hard to use because option names are displayed as they are used internally by Delphi, not as they are described in the help or shown in the option dialogs.

When you want to change a component (or group of components) into another (for example, all `TButtons` on the form into `TBitBtns`), the only thing that can save you from the endless clicking is *Replace Components*.

Set Tab Order allows you to set the tab order in an intuitive way: shift+click on the controls in the desired order, activate the expert and your work is done.

Source Export saves the source code to HTML or RTF formats, or to the clipboard, with syntax highlighting preserved. You can export the whole file, the portion that is selected in the IDE, or you can edit the shown source to customize the output. *To Do List* is a 'to do' organizer with three priority levels, largely surpassed by the similar functionality in the Delphi 5.

As you can see, there is much to choose from and surely enough for everybody. I can only advise you to download GExperts and give it a try: you won't be sorry. Remember: www.gexperts.org is the site!

Primoz Gabrijelcic (email him at gp@fab-online.com) is R&D Manager for F.A. Bernhardt GmbH.