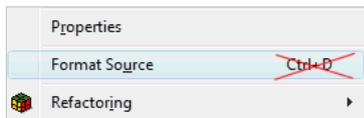


DDevExtensions 2.8 – Features

Disable Package Cache (default: off)

The RAD Studio IDE caches the component packages and their palette icons in the registry. This cache makes an IDE cold start slightly faster on some computers but has the drawback that the packages must be loaded later when the user opens a project or form. By disabling the cache the IDE start isn't harmed that much. Actually it could be even faster than with an enabled package cache, depending on your hard disks speed. (for SSDs disabling the package cache is recommended)

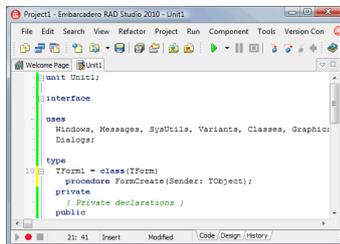
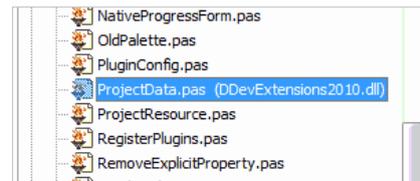


Disable Source Formatter hotkey (default: off)

RAD Studio 2010 introduced the Source Code Formatter. Unfortunately the hotkey that is used for it (Ctrl+D) can accidentally be pressed when you intend to press Ctrl+S (Save) or Ctrl+F (Find) in the standard keyboard bindings. DDevExtensions allows you to turn that hotkey off.

Show project for active file in Project Manager (default: on)

With DDevExtensions you can easily see to which project the current file in the editor belongs if there are multiple projects open.

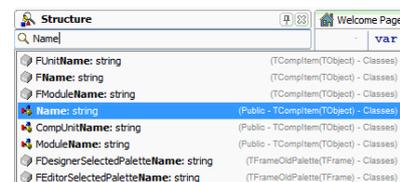


Editor tab double click action (default: zoom)

In Delphi 2007 there was a hidden registry key that allowed you to zoom or super-zoom the editor window by double clicking on the tab. DDevExtensions brings this feature back and also fixes some of the bugs that the original implementation had.

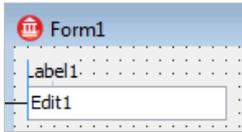
Structure View Search (default: no hotkey)

Browsing in the structure view is not that easy especially because the nodes always collapse if you switch to another view (form designer/code editor). The Structure View Search adds a search edit control above the structure view that allows partial search. That way you can find the location of a method or select a component by typing instead of mouse clicking. It is also possible to assign a hotkey that sets the focus to the Structure View Search.



Increment Build Number only when building the project (default: on) [2010 only]

RAD Studio 2010 changed the behaviour of the "Auto-increment build number" function to also increment the build number when compiling. DDevExtensions turns the functionality back to its original meaning. This also fixes the "Source has been modified. Rebuild?" bug that comes from the new behaviour because after compiling the project the build number is incremented what changes the project file and the debugger takes that as a source code change.



Set TLabel.Margins.Bottom to zero (default: on)

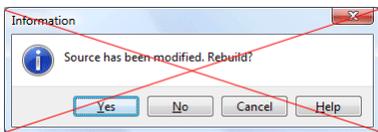
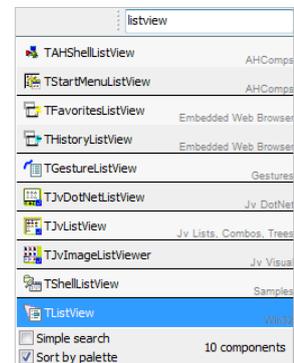
RAD Studio’s designer guide lines for TLabel force all controls to have a vertical distance of 6 pixels from a TLabel. This is because the TLabel bottom margin defaults to 3 and all other controls have a top margin of 3. The designer guide lines add both values together to get the distance between the controls. But 6 pixels let the two controls appear disconnected. DDevExtensions changes the default TLabel.Margins.Bottom value to zero.

Remove Explicit* properties (default: off)

The Explicit* properties prevent older Delphi/C++Builder versions from loading DFM file that were created by newer RAD Studio versions. With DDevExtensions you can automatically remove all Explicit* properties from the forms and frames that are modified and saved. It doesn’t run over the project.

Component Selector (default: off, no hotkey)

DDevExtensions adds a component search edit to the IDE’s toolbar. This search box allows you to filter the available components if you are in the form designer and it allows you to create new units, data modules, forms and derive from forms, frame and data modules if you are in the code editor. By selecting a component by mouse or pressing the ENTER key the component is added to the form or the action behind the list item is executed. It is possible to switch between partial search and simple search and a hotkey that sets the focus to the component selector control can be configured.



Disable “Source has been modified. Rebuild?” (default: on)

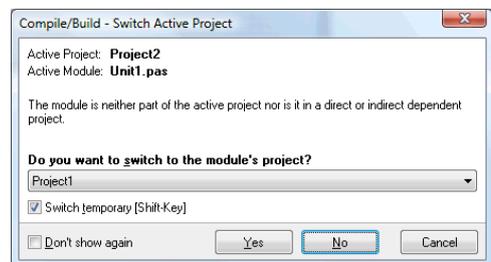
When debugging a project and you make changes to the source code intentionally or accidentally the IDE will ask you if you want to rebuild the project. DDevExtensions allows you to disable this annoying dialog.

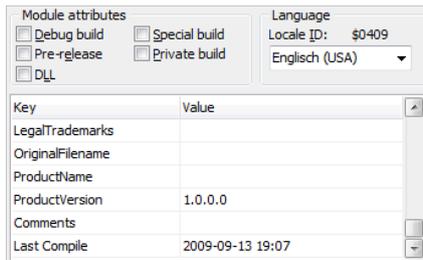
Auto-save editor files after successful compile (default: off)

DDevExtensions allows you to configure the IDE to save all modified files after a successful compile/build. This extends the IDE’s own auto-save option that only saves the files if you start the project.

Switch project to current file’s project (default: on)

When working with multiple projects you might end up compiling the wrong project if you make changes in a file that doesn’t belong to the active project and is not in a project that is a direct or indirect dependency of the active project. With DDevExtensions the IDE will ask you to switch the project.



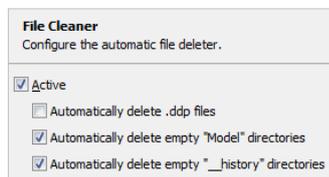
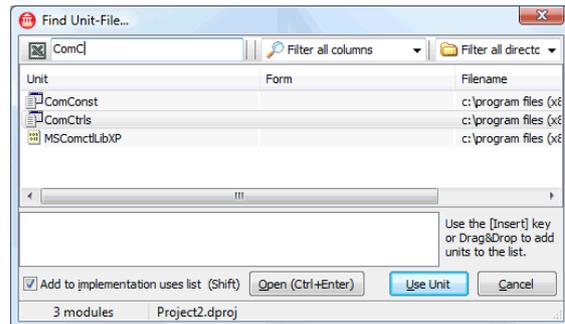


Put “Last Compile” time into version info (default: off) [2009-XE]

With DDevExtensions you can add the time stamp of the last compile/build into the project’s version info. The format of the time stamp can be configured in the DDevExtensions option dialog. The format follows the FormatDateTime() syntax.

Find Unit/Use Unit replacement dialog (default: on)

RAD Studio’s standard Use-Unit dialog only shows the files from the project group. DDevExtensions replaces the dialog with one that not only shows the project group files but also all files that the compiler can see. This makes it much easier to add units like Forms, Controls, ComCtrls or other library units that are not part of the project but in the global or project search path. In the new dialog you can select the units by using partial search and you can search for multiple units by separating the search masks by a space. Pattern matching with * and ? is supported. The dialog comes in two incarnations. As the Use-Unit dialog and as the Find Unit-File dialog that can be opened from the “Search” menu. For the later you can specify a hotkey in the DDevExtensions option dialog.

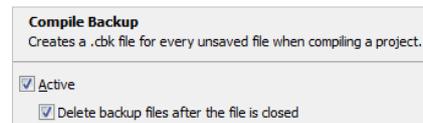


File Cleaner (default: on)

DDevExtensions File Cleaner deletes unnecessary files. It automatically deletes *.ddp files (older Delphi) and empty “Model” and “__history” directories.

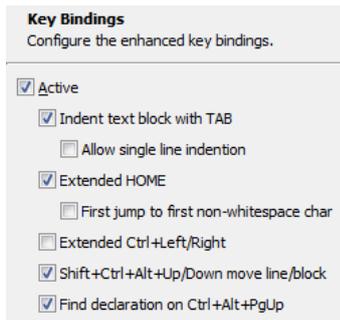
Compile Backup (default: on)

If you compile a project and you don’t like the auto-save function you can loose all your changes if the IDE crashes or even worse the computer crashes. Not so with the Compile Backup function. It saves all modified files as *.cbk files before the project is compiled. So if the IDE vanishes or the computer crashes you can restore the last compiled source code by removing the *.cbk file extension from the file names. The files are stored next to their base file and are automatically deleted if you close or save a modified file.



Loading package tweaks

If the IDE loads packages after the IDE start, e.g. if you install a package or you use the package cache, DDevExtensions will change the mouse cursor to crAppStart indicating the loading process and it also changes the IDE main form’s caption to include “[Loading package]”.



Enhanced Key Bindings (default: on)

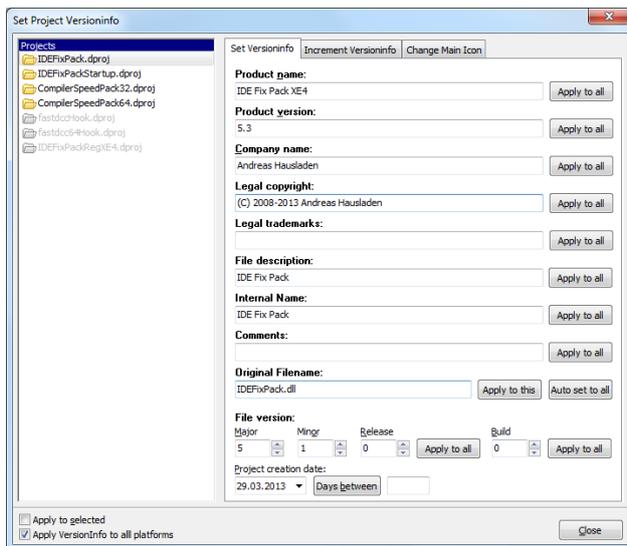
DDevExtensions changes the editor’s TAB key to indent or unindent the selected block like it is done in many other editors. Also the behaviour of the Home key is changed to jump to the first non-whitespace character if you press it twice. And if you are a friend of Visual Studio’s Ctrl+Left/Right movement then you can switch the “Extended Ctrl+Left/Right” option on. It also adds the hotkey Ctrl+Alt+PgUp for “Find Declaration”. For D2009 Shift+F3 for “invert find direction” is added.

Old Component Palette (default: off)

If you are a fan of the old Delphi 1-7 component palette you can switch on DDevExtensions “Old Palette” toolbar.



It is very limited in its features but supports multiline tabs, ragged right tabs, alpha sorted palette popup menu and small font tabs. It takes all the information from the IDE’s Tool Palette.

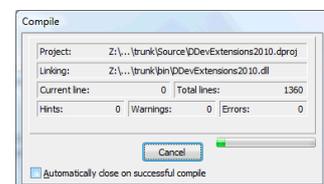


Set Project Versioninfo

The Set Project Versioninfo dialog makes updating the VersionInfo for multiple projects very easy. It has the ability to increment the version numbers instead of overwriting them. The dialog can be opened from the “Project” menu. The application Icon can be changed and it shows all icon formats that are in the *.ico file.

Compile Progress

DDevExtensions adds a progress bar to the compile dialog that indicates how many files of a project are already compiled. It is not 100% accurate, especially if your project uses many library units that are not listed in the project file. When the compilation has finished it shows the elapsed time. On Windows 7 and newer the progress is also visible in the taskbar button.



Replace “Package Add Unit” dialog with “File Open” dialog [2009 only]

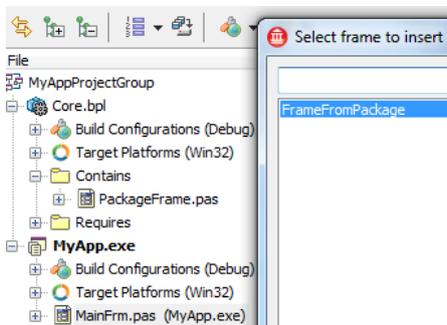
Delphi 2009’s “Package Add Unit” dialog automatically adds the file’s path to the global library search path. DDevExtensions replaces this dialog by the “File Open” dialog, like it is in Delphi 2010 and newer. This gets rid of the unwanted global library search path changes.

Close and Kill the IDE

You can force the IDE to terminate without executing all the destruction code if you press the Ctrl key when closing the IDE through the file menu or by pressing the window's close button. This causes the IDE to execute the close-code that saves the files and then it is terminated immediately without executing the destruction code. This is handy if your IDE throws exceptions if you close it. Furthermore DDevExtensions catches exceptions in the IDE's FormClose event and asks you if you want to kill the IDE. This makes it possible to terminate the IDE even if it can't be closed the normally.

Replace Open File At Cursor (default: off)

DDevExtensions can replace the IDE's "Open File At Cursor" functionality by a much more advanced implementation that supports package groups by not only looking in the project of the editor file, but also in the active project and all other opened projects.



Show all inheritable modules (default: off)

With the "Show all inheritable modules" feature it becomes possible to easily use visual inheritance and Frames that aren't in the current project but in a package project in the project group.

Don't break when starting spawned processes (default: off)

If the debugger is configured to debug into spawned processes, it stops the execution of the spawned process automatically and shows the CPU view. DDevExtensions can disable this hard breakpoint.

Disable Code folding (default: off) [2009 only]

DDevExtensions adds the missing "Disable Code folding" option for Delphi/C++Builder 2009 which can be necessary if you work with very large unit files.

Kill all dexplore.exe when closing the IDE (default: on)

Document Explorer (dexplore.exe) is used to show the context help but sometimes it isn't terminated what prevents Windows from shutting down. If this option is active, all running dexplore.exe processes will be killed when the IDE is closed.

Show confirmation dialog for Ctrl+F1 while debugging (default: on)

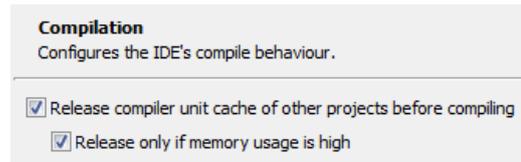
During a debug session it sometimes happens that you accidentally press Ctrl+F1 (Help) instead of Ctrl+F2 (Program Reset) and then you have to wait for the slow starting context help. This feature shows a confirmation dialog before invoking the help, allowing you to cancel the context help.

Disable Alpha-Sort Class Competition (default: off)

Delphi's class completion (Ctrl+Shift+C) inserts new methods in their alphabetical order. This features disables the alpha-sort so that the IDE appends the methods at the end of the class' method list like it was in older Delphi versions.

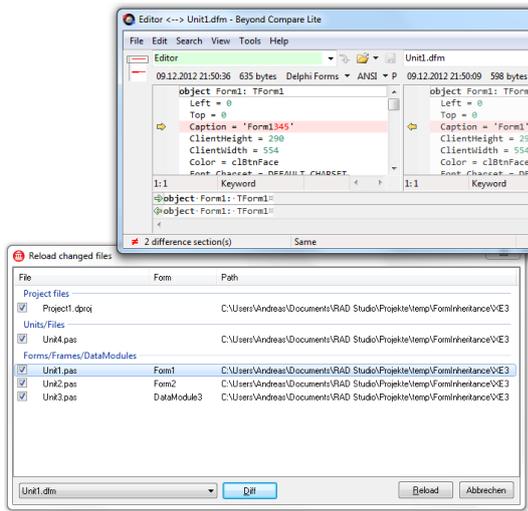
Release compiler unit cache before compiling (default: off, experimental)

If you compile large project groups the compiler’s unit cache can occupy a lot of memory for each compiled project. This can lead to “out of memory” exceptions. DDevExtensions can be set up to release the compiler’s unit caches of the projects that are already compiled. It is also possible to only release the caches if the IDE’s memory usage is high so CodeInsight doesn’t get a performance hit due to a released unit cache.



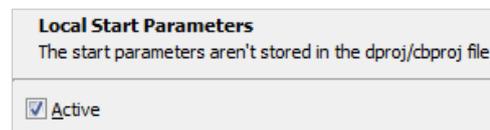
Improved reload changed files dialog

If a file that is opened in the IDE is changed in another application the IDE shows a reload file dialog. It is displayed for each modified file. DDevExtensions replaces this dialog with one that shows all changed files in a list, so you can reload or cancel all with one click. The dialog also supports diff-Tools like BeyondCompare (Lite) or TortoiseDiff which can show you the differences between the file in the editor and the file on disk. This comes in handy if you forgot to save your changes and did a svn update. You can then merge the unsaved changed into the file on disk and reload it.

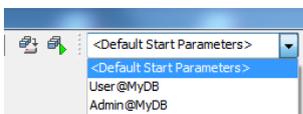


Local Start Parameters (default: off) [2009-XE]

If you work in a team and for example you use the start parameters to auto-login into a database and each team member must login to different databases the team members will have a hard time if the dproj/cbproj file is in a version control system. With Local Start Parameters the start parameters are extracted from the dproj/cbproj file and written into a *.projdata file that can be left out of the version control system. (See also “Project Start Parameters”)



Project Start Parameters (default: off)



Instead of setting the Start Parameters in the project options, what causes a recompile, you can choose between multiple configured Start Parameters from a ComboBox. The parameter sets are loaded from a <Project>.params XML file that can use macros and read lines from external files to construct the actual command line parameters.

It can be enabled by right clicking on the IDE’s toolbar and selecting “Start Parameters”. This will show the Start Parameters toolbar. If a project is open you can right click on the ComboBox and choose “Create <project>.params”. This will create the *.params XML file with an example and description comment next to your project’s *.dproj file. Once the *.params file is created you can edit it by right clicking on the ComboBox. You can also create a *.params.local file what is a replacement for the missing “Local Start Parameters” feature in XE2 and newer.

