

12 June 2007

By: Roxana Popa, Linux Editor



Ajuta logo

[Anjuta IDE Review](#)

C/C++ versatile IDE on GNU/Linux

The programming under Linux issue has always been a fashionable topic among different developers, mainly due to arguments regarding the IDEs (Integrated Development Environments) used for creating the applications. While some more "traditionalist" programmers still claim that the command-line GNU tools are by themselves an IDE, the graphical IDEs are gaining lot of popularity lately. There has been noticed even a tendency of combining the Open Source philosophy with an open, extensible framework, for creating a community of people to extend the capabilities of the IDE, adding even exotic languages and applications to the list of supported ones in the environment. Among the most popular IDEs there would be the KDevelop, an integrated development environment for the K Desktop Environment (with support of languages such as C++ or Ruby), Eclipse, NetBeans, or Anjuta, for C/C++. I'll tell you more over the later, as it happens to be a "friend" of mine (it is the IDE where I work on my projects). Anjuta is one the most advanced and sophisticated IDEs available for Linux, yet friendly and easy-to-use, and it is GPLed. It has been written for GTK+/GNOME and features a number of advanced programming facilities such as: a project management, some application wizards, an interactive debugger built over the GNU debugger (gdb) and a powerful source code editor, which offers you features like source browsing, code completion and least but not last I could add the syntax highlighting. As any regular IDE, Anjuta is made out of a source code editor, a compiler and/or interpreter, build-automation tools, and a debugger. It also integrates a class browser, an object inspector and a class hierarchy diagram, for use with object oriented software development. I happen to be a C/C++ programmer, therefore I am familiar with the C-ish side of Anjuta, which looks very good, I could add. I should definitely say a few words over the source code editor. What I've noticed in it is that it looks as simple as possible, but provides the exact functionality you need. Its most important, and useful, feature was the syntax highlighting. The newbies in the field will understand me better on this one. The editor can also be customized in order to provide certain information such as the indentation guides, white spaces and line end characters, for a more comfortable editing. In order to make these visible /invisible just go toggling the menu items in *View->Editor*. There are also features such as Finding/Replacing pieces of text in a file which increase the usefulness rate of this editor. And what kind of editor would Anjuta's be if it wouldn't support file printing? Whenever you need to print a file, just select *File -> Print Preview* to get a preview of what's going to be printed and then just press the *Print* button. Another aspect of the code editor that can be found in any respectable IDEs are the tabbed windows. On the compiler side, Anjuta, as most other IDEs, make use of the GCC for compilation. There is also *make* for project management and CVS for versioning. It also benefits of an on-board interactive debugger. Relatively small compared to Kdevelop, Anjuta stands out, as it is quite fast. I do not have to wait for ages for it to deal with my more-or-less buggy pieces of programs. Even though in the beginning it gained a pretty stained name due to the numerous bugs it had, now Anjuta comes with one of the most stable IDE versions. The fact that it looks a bit similar to the Visual Studio from Microsoft makes it highly accessible to the former Windows users, who were recently "enchanted" with Linux and look here for some more "familiar" stuff to use for their older activities. File compilation is made easy by simply pressing the **F9** key, or by selecting *Build -> Compile*. You will receive in the Message window a small compilation log, including possible errors. You can double-click on the error lines which will take you to the particular line where the error occurred. The ordinary *Compile* option will not build you the executable. For that, you can press the **F11** key (or following the *Build -> Build*

). You will receive a message list in the Messages window here too. Anjuta has also a smooth way of maintaining and managing projects. It also allows you to make use of other tools too, which have an additional advantage of making your projects more flexible. You'll have to take care though with the files managed by Anjuta, as they would all be overwritten.

The Good Looks good, it is relatively light, it does lots of things automatically (such as Makefiles or configure scripts) and benefits of an extensive documentation. And these are just a few of Anjuta's capabilities.

The Bad Anjuta is a good, flexible, stable and reliable development environment. That's for sure. It has a nice way of managing projects, a very useful and advanced source code editor and a pretty good interactive debugger. But (you know ... there's always a but) it still needs improvements in many domains. Let's take the editor for example. I've said before it deals nicely with windows as it has the tabbed window option. So far so good. When you open the file, let's say a .cc file, the application does not create a net tab for that certain file, but instead runs another instance of Anjuta. Another thing I did not like about it was the Terminal. Unfortunately, this incorporated terminal proposed by Anjuta does not support copy-pasting. I hope this will be solved in the future releases.

The Truth Anjuta is an Integrated Development Environment for the C and C++ programming languages on GNU/Linux. It has been written for GTK/GNOME and features a number of advanced programming facilities. Among these, there would be: project management, application wizards, an on-board interactive debugger. *If interested, you can also take a look at the Anjuta photos below:*