

<u>wxDev-C++</u> is an extension of <u>Dev-C++</u> by Colin Laplace *et. al.* This program helps you to visually create dialogs and frames for <u>wxWidgets</u>. With all the wonderful features of Dev-C++, wxDev-C++ is still being actively developed. The main aim of this project is to provide the wxWidgets community with a free, open-source, commercial-grade <u>IDE/RAD</u> tool for development with wxWidgets.

Features:

Plugin Capable:

- Modules can be added at runtime to expand the IDE's capabilities
- Users can chose between wxDev-C++ and the original Dev-C++ IDEs

wxWidgets Form Designer

- Generates XRC XML resources
- Drag-and-drop design paradigm
- Supports wxWidgets' sizer-based layouts
- Connect events to member functions within the editor

Integrated debugging

- Support for GDB and CDB (WinDbg)
- Variable watches
- Automatic stack tracing
- Local variables list
- Displays disassembly and CPU registers

Editor features

- Class browser
- Code completion
- Project management
- Project profiles
- Customisable syntax highlighting
- Automatic inline assembly highlighting
- ToDo List

Application compatibility

- Built-in CVS support
- Supports MingW/Visual C++ (6, 2003 and 2005, 2008)

Quickly create Windows and console applications, static libraries and DLL's

Support for project templates to expedite the creation of new project types

Package manager (through the use of DevPaks), for easy installation of addon libraries < Previous < = Home = >Next >

License Agreement

Dev-C++ and wxDev-C++ are distributed under the <u>GNU General</u> <u>Public License</u>.

Be sure to read it before use.

GNU GENERAL PUBLIC LICENSE Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the

Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN

OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES

PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED

OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF

MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS

TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE

PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING,

REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING **BUT NOT LIMITED** TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO **OPERATE WITH ANY OTHER** PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN **ADVISED OF THE**

POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the

GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.

This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be

mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989 Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. < Previous < = Home = >Next >

Credits

For <u>Dev-C++</u>

Developers : Colin Laplace, Hongli Lai, Mike Berg, Yiannis Mandravellos

<u>MingW</u> compiler system : Mumit Khan, J.J. Var Der Heidjen, Colin Hendrix and GNU developers

Update system and initial work on the help file : Kip Warner

New Look theme : Gerard Caulfield:

Gnome icons : Gnome designers

Blue theme : Thomas Thron

For <u>wxDev-C++</u>

Developers: Guru Kathiresan, Malcolm Nealon, Esteban Aguilar, Edward Toovey, Nuklear Zelph, Joel Low, and Tony Reina

Update of Help File: Tony Reina

< Previous < = Home = >Next >

Getting Started

When you launch Dev-C++ for the first time, you will see the following dialog :

File Edit Search Image: Search This is the first time you have launched wxDev-C++. Here are some questions to help configure wxDev-C++ to your liking. Image: Select your language : Chinese (TW) Image: Select your language : Chinese (TW) Image: Select your language : Creatian Image: Select your language : Image: Select your language : Image: Select your language : Image: Select your language : Image: Select your language : Image: Select your language : Image: Select your language : Image: Select your language : Image: Select your language : Image: Select your language : Image: Select your language : Image: Select your language : Image: Select your wxDev-C++ theme : Select your wxDev-C++ theme : Image: Select your wxDev-C++ theme : Select your wxDev-C++ theme : Image: Select your wxDev-C++ theme : Image: Select your wxDev-C++ theme : Image: Select your wxDev-C++ theme : Image: Select your wxDev-C++ theme : Image: Select your wxDev-C++ theme : Image: Select your wxDev-C++ theme : Image: Select your wxDev-C++ theme : Image: Select your wxDev-C++ theme : Image: Select your your your your your your your your	wxDev-C++ First-Run Con	figuration
Image: Construction of the second	File Edit Search Image: Image of the search Image of the search	This is the first time you have launched wxDev-C++. Here are some questions to help configure wxDev-C++ to your liking. Select your language : Chinese (TW)
8 // □ Use XP Theme 9 // ↓ □ Use XP Theme	Property dummy.cpp app.cj 1 // 2 // 2 // 3 // Nar 3 // Nar 4 // Aut 5 // Cre 7 // Ret 7 // Ret 8 // 9 / 10 11 #inclt	Croatian Czech (Èeština) Danish Dutch (Nederlands) English (Original) Select your wxDev-C++ theme : New Look <u>Preview</u> <u>Use XP Theme</u>

You can select the language wxDev-C++ will use, as well as the icon theme.

To add a XP flavor to wxDev-C++, check the 'Use XP Theme' box.

If you installed wxDev-C++ with the included Mingw compiler system, then all MingW compiler paths will be set to their correct values. If you have previously installed versions of Cygwin, MSYS, or MingW, they might interfere with wxDev-C++. Although they can be used, you'll have to set up the compiler paths and settings manually in the *Tools* menu under *Compiler Options*.

If you also have the MS Visual Studio compiler installed, wxDev-C++ will attempt to detect it and add the relevant paths as well.

Now you can then proceed to the <u>Basic Steps</u>.

< Previous < = Home = >Next >

Creating a Project

What is a wxDev-C++ Project ?

Projects are the way you manage different source files and compiler/linker options inside wxDev-C++. It's an ini file (usually ending in the *.dev* extension), which specifies what source files should be included in your program and what compiler/linker parameters/switches you want to use to build them.

When to use wxDev-C++ Projects ?

(Answer: [Almost] Always)

- *You have more than one source file* Using Projects ensures that all of your source files are linked to the executable after they are compiled.
- You need to create a DLL or static library, or want to use resource files in your program Projects allow you to specify the compiler and linker options necessary

How can I create a wxDev-C++ Project ?

Go to the *File* menu and click on *New*, then *Project*. A dialog opens, containing different Project types.

ew project						X
Basic Intro	duction Mul	tiMedia				
Empty wxWidge Description:	wxWidgets Dialog	wxWidgets Frame	Console Application	Static Library	de de la compañía de	
Project optio Name: Project1	ns:			<u>C</u> Project <u>M</u> ake Default L	¯⊙ C <u>+</u> + Proj .anguage	ect
				k XC	ancel	P <u>H</u> elp

Here are the basic Dev-C++ project types :

- Console application : creates a console (command-line shell) program
- **Static library** : creates an empty project with the options needed for building a static library
- **DLL** : creates a Win32 Dynamic Link Library

In addition, wxDev-C++ contains three wxWidgets (GUI) project types:

- **Empty wxWidgets project** A project that initially contains no source files but has the correct compiler and linker settings to build wxWidgets source code.
- **wxWidgets Dialog** A project that contains skeleton source files for a wxDialog
- **wxWidgets Frame** A project that contains skeleton source files for a wxFrame

Now that you selected your project type, fill the name of your project in the corresponding edit box, select the programming language you will be using (C or C++), and click on OK.

< Previous < = Home = >Next >

Adding and Removing files

Adding and removing files is very easy.

You can add multiple files in the same time by clicking on *Project* menu, then on *Add to Project*.



You have different ways for removing files from your project.

Either click on *Project* menu, then on *Remove from Project* and select the file you want to remove in the list, or right-click on the file you want to remove in the **Project Manager** window, and click on *Remove file*

< Previous < = Home = >Next >

Project options overview

You can load the Project Options dialog by clicking on the *Project* menu, then on *Options* (shortcut :**Alt+P**).

General sheet :

eneral Files	Compiler	Parameters	Directories	Build Options	Makefile	Version info
lame:	Project1					
ilename:	C:\Users\tony	\Documents	Programmin	ig\junk\Project1	.dev	
Output file:	C:\Users\tony	\Documents	Programmin	ig\junk\Output\	MingW\Pr	oject1.exe
iles:	6 files [2 source	es, 2 headers	, 1 resources]			
Icon:			Type:			
	::: Libr	ary	Windows GL	II Application		<u>^</u>
	Broy	NSE	Windows Co Windows Sta	nsole Applicatio	on	
			Windows Dy	namic Library		-
1	Rem	ove	Support W	indows XP Ther	nes	

The first thing you may notice is that wxDev-C++ allows for <u>multiple</u> <u>compiler profiles</u> for the same project. In the **Project Profile** section there is a drop-down box with the available compiler profiles for the current project. Users can switch the compiler profile between different compilers (e.g. MingW gcc and VS 2008) or between different settings of the same compiler (e.g. debug versus release profiles). You may add, remove, rename, or copy the profiles for the current project by clicking on the icons to the right of the drop-down box.

I \blacksquare \blacksquare (Note that these actions only affect the current project.)

Other items of this window:

Name: Modify here the name of your project.

Icon : You can assign an icon to your program, either by selecting one in the Icon Library, or by giving your own icon using the Browse button.

Type : This is an important settings which indicates which project type you are making. Select :

- Win32 GUI : if your application is a graphical user interface
- **Win32 Console** : if your application needs a console window (MS-DOS or command shell window)
- Win32 Static Lib : if you are creating a static library
- Win32 DLL : if you are creating a dynamic link library (DLL)

Files sheet :

eneral	Files	Compiler	Parameters	Directories	Build Options	Makefile	Version info
-	Proiect1			File optio	ons:		
- -	Proje	ct1App.cpp		Build prid	ority: 0		
	Proje	ct1App.h		Includ	le in compilation	n	
	🔒 Proje	ct1App.rc		Includ	le in linking		
	😑 Proje	ct1Frm.cpp		Comp	ile file as C++		
	🔒 Proje	ct1Frm.h		Overri	de build comm	and	
	😑 Proje	ct1Frm.wxfo	orm				~
							-
				٠			P.
				Precomp	iled Headers		
				No pr	ecompiled head	ler	
				Create	e precompiled h	eader	
				O Use pr	recompiled head	der	

Regardless of the compiler used, wxDev-C++ uses the MingW make build

system to compile and link projects. Makefiles are automatically generated based on the source files contained in the project. This window enables you to modify the compilation commands and options for each source file to customize the makefile that is generated.

Build priority : Increment this value in order to have the source file compiled in priority of the others

Include in compilation : If not set, your file will not be compiled.

Include in linking : Add the object file generated from the source file to the linking stage

Compile file as C++ : Check this flag if it is a C++ source file

Override build command : For experimented users only. You can change there the command used by Dev-C++ to compile your file

Compiler sheet :

Project	Options							X
Project Pr	ofile: Mi	ingW 3.4.2				•	+ 🗙 (1
General	Files	Compiler	Parameters	Directories	Build Options	Makefile	Version i	nfo
NOTE: T	hese sett	tings will ove	erride the glot	oal Compiler	Options affectin	ig this proje	ect only.	
Compile	er: Def	ault GCC co	mpiler					•
C + C+ Co Lin Op	compiler: + compil de Gener de profili ker: timizatio	ler: / ration I ng (Support all AN Attempt to su Inhibit all warı Displays one e	ISI standard (pport some a ning message error per line	C programs ispects of traditi es	onal C prep	No No No	•
					🗸 <u>O</u> k 🚺 🍞	Cancel	?	<u>H</u> elp

See the Compiler Options section for more details.

Parameters sheet :

eneral Files Compile	er Parameters	Directories	Build Op	otions	Makefile	Version info
Compiler:	C++ co	mpiler:		Linker	r:	
-fno-exceptions -fno-pcc-struct-return -fstrict-aliasing -Wall -D_WXMSW_ -D_GNUWIN32_ -D_WIN95_ 		cceptions cc-struct-retu -aliasing KMSW NUWIN32 N95	rn E	-mw -lwxr -lwxr -lwxg -lwxg -lwxg -lwxz -lwxr -lwxr -lwxr -lwxr -lwxr	indows msw28 msw28_gl iiff peg png elib regex expat nel32	
WXMSW _GNUWIN32_ _WIN95_			*	-luse -lgdi ∢	er32 32	

You can provide here command line arguments to the C/C++ compilers and linker.

Use the linker parameters box to specify libraries to link with your project. For more information, please read the linker library section.

Directories sheet :

957	Project	Options	1000						
P	roject Pr	ofile: Mir	ngW 3.4.2			1	-	+ ×(IL
	General	Files	Compiler	Parameters	Directories	Build Options	Makefile	Version i	nfo
Ľ	Library	Directorie	s Include	Directories	Resource Dire	ectories			
									0.50
									•
		F	Replace	Add	Delete	Delete Inv	alid		
						🗸 <u>O</u> k 📄 🏹	Cancel	?	<u>H</u> elp

You can provide here a list of Includes, Resources and Libraries directories to be searched when compiling/linking. These directories are only used for the current project and profile.

Build Options sheet :

		- -	D	Di la i	Ruild Ontions	MI CI	
eneral	Files	Compiler	Parameters	Directories	Build Options	Makefile	Version into
Directo	ories						
Execut	able out	put director	/				
Outp	ut\Ming\	N					
Ohian	file aut	aut disastan					
Object	tel Minel	out directory					
Objec	ts\iving	VV);=
Over	ride outp	ut filename					
Droi	act1 eve						
rioj	L'ULICAL						

Executable output directory : Specify here the directory where your executable will be created (default is project's directory).

Object file output directory : Specify here the directory where your object files will be created (default is source file's directory).

Override output filename : You may change the output filename of your program here.

Makefile sheet :

eneral Files Compiler Paramet	ters Directories	Build Options	Makefile	Version info	
Use custom Makefile (do not gene	rate a Makefile, u	use this one)			_
Customizing the Makefile The makefile has two main targets, 'all' and 'clean'. 'all' depends on all-before and all-after. all-before and all-after gets called before and after the compilation process respectively. 'clean' depends on the target	Include the f	ollowing files into	the Makef	ile	
clean-custom, which gets called before the cleaning process. Alter the Makefile's behavior by defining the targets mentioned.					

wxDev-C++ automatically creates a MingW Makefile for taking care of the building process. If you are experienced with makefiles and want to add other lines, you can do it here. Or, you can create your own custom makefile and have the IDE use the custom makefile rather than the auto-generated one.

Version Info sheet :

eneral Files	Compiler	Parameters	Directories	Build Options	Makefile	Version info)
Traduda ur						1	
Version deta	ails	ion in project	Additio	nal information			
Major	0		Eile De	ention			
Minan	1		File Ve	rsion			
winor:			Produ	ct Name			
Release:			Produ	ct Version			
Build:	1		Origin	al Filename			
Language:	English (Unit	ed States) 🔹	Interna	al Name			
			Comp	any Name			
Build numb	er increment		Legal	Copyright			
Don't Au	sto-Increment	build number	Legal	rademarks			
O Auto-Inc	crement build r	number on reb					

You can specify version information for your program here. This info shows up when you right click on the executable's icon and select "Properties".

< Previous < = Home = >Next >

Linking libraries with your project

A little history :

MingW/GNU

Library filenames under the GNU system are in the form *libNAME.a* (where NAME is the name of the library, like wsock32).

For example, if you want to use the wsock32 (winsock) library, the filename will be *libwsock32.a* The gcc parameter for linking a library is –*NAME*, so for linking with the wsock32 library we would give GCC the –lwsock32 parameter.

MS VC++

MS libraries are in the form *NAME.lib* (where NAME is the name of the library, like wxmsw28). MS VC uses the entire name so, for example, linking the wxWidgets 2.8.9 library would be wxmsw28.lib.

Linking your library:

Click on Project menu then on Options. Now click on the Parameters sheet.

General Files Compiler	Parameters	Directories	Build Op	tions	Makefile	Version i	nfo
Compiler:	C++ cor	npiler:		Linke	r:		
-fno-exceptions -fno-pcc-struct-return -fstrict-aliasing -Wall -D_WXMSW	 -fno-ex -fno-po -fstrict- -Wall -D_WX -D_GN -D_WII 	ceptions :c-struct-retu aliasing :MSW UWIN32 N95	rn E	-mw -lwxi -lwxi -lwxi -lwxi -lwxi -lwxi -lwxi -lwxi -lwxi	rindows msw28 msw28_gl tiff jpeg png zlib regex expat nel32 ar32		
WXMSW GNUWIN32 WIN95			+	-lgdi ∢	i32	Þ	-

In the *Linker* edit box, you can specify as many libraries as you need. You can also pass the complete filename of the library.

Example : -lm -lwsock32 c:\libs\mylib.a c:\objs\myobj.o

You may also use the *Add Library or Object* button Add Library or Object to select your library from a list.
Compiling and linking process

How does the compile and link process works ?

The build process can be generally divided into four steps :

- 1. *Preprocessor* : expands macros and include files in your source.
- 2. *Compiler* : transform your source file into assembly code (a processor language that's human readable)
- 3. *Assembler* : takes the assembly code and generate machine-readable code (binary object code)
- 4. *Linker* : assembles and resolves object codes together to create a single executable.

How to do this in wxDev-C++ ?

Just go to the *Execute* menu and click *Compile* (shortcut : **Ctrl+F9**), and wxDev-C++ will take care of the 4 build steps.

Look at the bottom panel of wxDev-C++, you should get something like this :

Compile Log		₽ ×
Compiler 🙀 Resources 🚮 Com	npile Log 🛛 🗔 Debug 🛛 🔎 Find Results 🛛 🍓 To-Do List 🛛	
Information:	Compile Log:	
Total Errors:	Compiler: Default GCC compiler	*
Size of Output 3281641 butes (3204 K	Building Makefile: "C:\Users\tony\Documents\Programming\junk\Makefile.win"	
Size of Output. 5201041 bytes (5204 K	Executing make clean	
Abort	m -f Objects/MingW/Project1App.o Objects/MingW/Project1Frm.o Objects/MingW/Project1_private.res	
	Output/MingW/Project1.exe	
	g++.exe -c Project1App.cpp -o Objects/MingW/Project1App.o -I"C:/Program Files/Dev-Cpp/include" -	
	I"C:/Program Files/Dev-Cpp/" -I"C:/Program Files/Dev-Cpp/include/3rdparty/wx/plotctrl" -I"C:/Program Files/Dev	-
54:1 Cor	npilation successful; Compilation took 10 seconds	

This log window shows you what wxDev-C++ is doing (the program it executes, the files it is creating, ...).

If your program compiled, you will see the message 'Compilation successful' in this window.

If compiling or linking fails, you will see a list of the errors on the bottom panel :

iller 🙀 Resources 📶 Compile Log 🔯 Debug 🔎 Find Re	sculte 1 🎭 To Do Litt 1	
ile	Message	
C:\Users\tony\Documents\Programming\junk\Project1Frm.cpp	In member function 'void Project1Frm::CreateGUIControls()':	
C:\Users\tony\Documents\Programming\junk\Project1Frm.cpp	i' was not declared in this scope	
C:\Users\tony\Documents\Programming\junk\Project1Frm.cpp	[Warning] unused variable 'i'	
C:\Users\tony\Documents\Programming\junk\Makefile.win	[Build Error] exe: *** [Objects/MingW/Project1Frm.o] Error 1	
Insert 89 Lines in file		1
	le \\Users\tony\Documents\Programming\junk\Project1Frm.cpp \\Users\tony\Documents\Programming\junk\Project1Frm.cpp \\Users\tony\Documents\Programming\junk\Project1Frm.cpp \Users\tony\Documents\Programming\junk\Makefile.win Insert 89 Lines in file	In member function `void Project1Frm::CreateGUIControls()': `\Users\tony\Documents\Programming\junk\Project1Frm.cpp In member function `void Project1Frm::CreateGUIControls()': `\Users\tony\Documents\Programming\junk\Project1Frm.cpp `` was not declared in this scope \Users\tony\Documents\Programming\junk\Project1Frm.cpp `` was not declared in this scope \Users\tony\Documents\Programming\junk\Makefile.win [Warning] unused variable 'i' [Build Error] exe: *** [Objects/MingW/Project1Frm.o] Error 1 Insert 89 Lines in file

You can directly jump to the line in your code where the error appears by double-clicking on item.

Note, the GNU/MingW Makefile system that we use should detect any files that have been changed between builds and only re-compile those files. However, there are some times when you may want to just force the Makefile system to rebuild everything in the project. To do this, just go to the *Execute* menu and *Rebuild All* (shortcut: **Ctrl+F11**). Rebuild all is equivalent to calling *Clean* and then *Compile*.

Executing your program

The Basics

Executing your program is as simple as clicking on the *Execute* menu, then *Run* (shortcut : **Ctrl+F10**).

You can also use *Compile and Run* ³ (shortcut : **F9**) to build your program and then executing it.

Introduction to debugging

All developers make mistakes, but some mistakes are harder to detect than others. Compile-time mistakes are usually the easiest to spot because wxDev-C++ will provide a list of those errors (including line numbers and suggested corrections) in the *Compiler Output* window. Runtime errors (or "bugs") are more insidious because the developer has no way of knowing what line of code caused the dreaded infinite loop or "Access Violation" or "Segmentation Fault" or "Blue Screen of Death". Thus debuggers were created to help developers investigate their program while it is running.

A debugger is a program that runs your program inside of it. It keeps track of your program's functions, variables and instructions. It is capable of pausing your program at a given moment (aka breakpointing), allowing you to view (and even modify) the values of your variables at that moment, and then continuing the execution of the program either one instruction at a time (stepping) or to the end of execution. You can set breakpoints anywhere in your code : once your program reaches that code at runtime, the debugger will pause your program and let you examine its variables at that time.

This tutorial explains the use of the MingW GNU debugger (gdb). It can <u>only</u> be used with the MingW gcc compiler. Note that gdb is intended to be a command-line interface. It runs as its own shell allowing the user to run gcc-compiled programs from within that shell. All we do in wxDev-C++ is send messages to the gdb shell and parse the output as it occurs.

With gdb (as with all debuggers), we can:

- specify places within the program to halt execution (aka breakpoints)
- step through the program one line of source code at a time
- view the values of variables in realtime as the program executes
- evaluate the memory stack when a program throws an error

Note that a web search will reveal many gdb tutorials out there that are far more comprehensive than one in terms of scope (for example, <u>http://dirac.org/linux/gdb/</u>). Please consider and review them if yo want to learn the in's and out's of using gdb.

There are, in fact, other graphical programs, which also use gdb in a similar manner. For instance, th Display Debugger (<u>ddd</u>) is a popular GUI interface for gdb debugging. You may, in fact, prefer to use 3rd party programs instead of the built in wxDev-C++ interface.

Example Program to Debug

We're going to explain the debug process by working through an example. Although we are limiting our discussion to the gdb debugger, the same example could be used for other debuggers.

To create our example debug project, go to the *File* menu and select *New* and then *Project*. The new project dialog will be displayed.

ew project		Ster.				X
Basic Intro	duction Mul	ltiMedia				
Empty wxWidge Description:	wxWidgets Dialog	wxWidgets Frame	Console Application	Static Library	dll	
Project optio Name: Project1	ns:			<u>C</u> Project <u>M</u> ake Default La		ct
				Can	ncel ?	<u>H</u> elp

Select the **Console Application** from the window. We'll name our project "sampleDebug". You'll notice that a new project will be created with a skeleton C++ code called **main.cpp**. Replace the C++ code in **main.cpp** with the following code:

	main.cpp
#include <cstdlib> #include <iostream></iostream></cstdlib>	
using namespace std;	

```
float fGlobal = 1234.56;
void test2(int* iTest2a, int iTest2b)
{
    char chTest2 = 'r';
    *iTest2a = iTest2b;
    printf("Finished test2\n");
void test()
    int iTesta, *iTestb; /* Put a breakpoint here */
    test2(&iTesta, 5);
    printf("iTesta = %d\n", iTesta);
    /* The next lines will cause a runtime error
               since iTestb never gets initialized */
    /*
    test2(iTestb, 3);
    printf("iTestb = %d\n", iTestb);
    */
int main(int argc, char *argv[])
{
    float fMain = 3.1415;
    test();
    printf("Press any key to continue...");
    getchar (); /* Pause the program from exiting */
    return EXIT_SUCCESS;
```

Remember to save the project after you've replaced the contents of **main.cpp**.

This sample project will demonstrate the concepts of functions, local and global variables, breakpoints, and backtraces. Note that lines 24-27 are currently commented.

```
/*
test2(iTestb, 3);
printf("iTestb = %d\n", iTestb);
*/
```

These lines will not cause an error during compile-time (i.e. the Mingw gcc compiler will create an executable with no reported compile errors), but it will cause a core dump at runtime due to the use of the uninitialized variable iTestb.

In the next few sections, we'll put our sampleDebug project to good use explaining how to use the debugger...

Debugging your program

Launching your program into the debugger is easy. Just go to the *Debug* menu and click on *Debug* (shortcut : **F8**).

If you do not have debugging information set in your project, wxDev-C++ will ask you if you want to rebuild your program with this information enabled. You can manually select that option in *Compiler Option* in the *Linker* section.

Project Options									
Project Profile: Profile 1					•	+ו•			
General Files Compiler	Parameters	Directories	Build Options	Makefile	Version in	nfo			
NOTE: These settings will override the global Compiler Options affecting this project only. Compiler: Default GCC compiler									
C compiler:	Link an Obj	jective C pro	gram prmation			No ir/es			
Code Generation Code profiling	Code profiling Strip executable No								
					Cancel	? Help			

Note that the executable size has grown because the compiler has added additional information that the debugger accesses during runtime. After your project has been rebuilt with the debugger information included, you can click *Debug* again.

The debugger has now loaded your program and runs it.

Debug	X
👷 Compiler 📶 Compile Log 🛛 🔯 Debug 🔎 Find Results 🔧 To-Do List	
Backtrace Local Variables Threads Watches Output	
Send command to GDB : Send	
<pre>Output to Display "Reading symbols from C:\\\\Documents and Settings\\\\R\\\\My Documents\\\\C++\\\\sampleDebug\\\\Output\\\\MingW\\\\sampleDebug.exe" Output to Display "done.\n" Output to Display "[New Thread 1940.0x17c]\n" Current Thread 1D = 1 Thread: id="1",target-id="Thread 1940.0x17c",frame={level="0",addr="0x00401453",func="main",args= [{name="argc",value="1"},{name="argv",value="0x32c90"}],file="main.cpp",fullname="C:/Documents and Settings/R/My Documents/C++/sampleDebug/main.cpp",line="34"},state="stopped" Thread 1 Stopped - Exited normally Debugger closed.</pre>	
	\mathbf{v}

What happened? Well, as the **Debugger Output** window shows above, wxDev-C++ started gdb and ran your executable called "sampleDebug.exe". The program had no errors and no breakpoints and so it exited normally. Since the program exited normally, this output is probably only interesting if you

are familiar with gdb.

You will also have seen your program run in a console window like this



You can also send commands directly to the debugger by using the "*Send command to GDB*" edit box just above the output. If you do not know gdb, you can type **help** to display a list of commands.

You can see (almost) all of GDB's output by turning on Verbose Debugger Output in the Debug menu.

Remember that wxDev-C++ is really just running a shell that passes these commands to the debugger and reads the output returned. You could re-create the same commands by using gdb from within a DOS shell.

In the next sections, we'll cover several useful (and more interesting) aspects of the debugger :

- Breakpoints
- Stepping through your code
- Watch variables
- Backtracing
- Using the CPU Window

Setting Breakpoints

You can use breakpoints to pause your program at a certain instruction (i.e. at a line of code).

To add a breakpoint, first select the line of code where you want to pause by simply positioning the text cursor on it. Now, click on the *Debug* menu, then on *Toggle Breakpoint* (shortcut : **Ctrl+F5**). Clicking on the gutter (at the left of the editor) in front of your line will have the same effect. You can set multiple breakpoints. Clicking on the checkmark in the gutter will remove the breakpoint.

Note that if the breakpoint is set, the line will be highlighted red and a checkmark will appear in the left gutter.

```
main.cpp
     6 float fGlobal = 1234.56;
                                                                       ×.
     7
    8 void test2(int* iTest2a, int iTest2b)
    9 {
    10
          char chTest2 = 'r';
    11
          *iTest2a = iTest2b;
          printf("Finished test2\n");
    12
    13 }
    14
    15 void test()
    16 {
   17 int iTesta, *iTestb; /* Put a breakpoint here */
    18
    19
         test2(&iTesta, 5);
    20
          printf("iTesta = %d\n", iTesta);
    21
    22
          /* The next lines will cause a runtime error
    23
                    since iTestb never gets initialized */
          /*
    24
    25
          test2(iTestb, 3);
    26
          printf("iTestb = %d\n", iTestb);
```

Now re-run the debugger (shortcut: **F8**). wxDev-C++ will run your program via the debugger and will warn you that your breakpoint was reached by changing the line color to blue.

```
main.cpp
     6 float fGlobal = 1234.56;
                                                                       .
     7
     8 void test2(int* iTest2a, int iTest2b)
    9 {
    10
          char chTest2 = 'r';
    11
          *iTest2a = iTest2b;
    12
         printf("Finished test2\n");
    13 }
                                                                       Ξ
    14
    15 void test()
    16 {
          int iTesta, *iTestb; /* Put a breakpoint here */
 🧉 17
    18
    19
          test2(&iTesta, 5);
    20
         printf("iTesta = %d\n", iTesta);
    21
    22
          /* The next lines will cause a runtime error
    23
                    since iTestb never gets initialized */
    24
          /*
    25
          test2(iTestb, 3);
    26
          printf("iTestb = %d\n", iTestb);
```

In the next steps we'll see how to examine the variables and step through the program's execution.

Stepping Through Your Program

Once a breakpoint has been reached, you can step into the code of your application in different ways :

- **Next Step** [I] (shortcut : **F7**) : The debugger will execute the next instruction (i.e. line of code) and pause
- **Step Into** № (shortcut : **Shift+F7**) : The debugger will execute the next instruction (i.e. line of code) and pause. If that instruction is a function call it will jump into the function and future steps will go line by line through that function until it returns to the line that called it.
- **-exec-finish** : The debugger will resume execution until the current function is exited.
- **Debug** / **Continue** (shortcut : **F8**) : The debugger will start or continue the execution of your program until another breakpoint is reached.

If your program is still paused at the breakpoint, try hitting **F7** to continue to the next step. You'll note that the cursor in the source code moves down to the next line of code and highlights it in blue. The breakpoint returns to its red highlight.

```
main.cpp
    12
           printf("Finished test2\n");
                                                                          ٠
    13 }
    14
    15 void test()
    16 {
   17
          int iTesta, *iTestb; /* Put a breakpoint here */
 ø
    18
    19
           test2(&iTesta, 5);
   20
           printf("iTesta = %d\n", iTesta);
    21
    22
           /* The next lines will cause a runtime error
    23
                   since iTestb never gets initialized */
    24
           /*
    25
           test2(iTestb, 3);
    26
           printf("iTestb = %d\n", iTestb);
    27
           */
    28 }
    20
```

If you keep hitting F7, the debugger will continue executing the program one line at a time until it reaches the end. Note that this is very useful in sections of your code where you think there may be logic errors or infinite loops. You can essentially slow the program execution down and view each instruction as it happens.

Watching variables

One of the most interesting aspects of debugging is the possibility to display the value of your variables at a given time. This way you can be sure your variable (e.g. a counter index) has the value you would expect.

wxDev-C++ is able to conveniently show you the contents of your classes, strings, structures/unions, arrays and other variables in the *Debug Local Variables* window:

Debug		×
Compile	er 📶 Compile Log 🛛 🔯 Debug 🖉 Find Results 🍓 To-Do List	
Backtrace	Local Variables Threads Watches Output	
Name	Value	Location
iTesta iTestb	"5" "0x77c35c94"	

Note that in the above window, iTesta has already had its value set to "5", but iTestb has not been set (the value will be random and depends on what data is at that pointer at that given time).

Now keep this *Local Variables* window open and re-start the debugging. You should be able to "step through" the code after the breakpoint (discussed in the <u>last section</u>) and watch the iTesta variable change from a random value to "5".

Adding Watchpoints

Watchpoints are like breakpoints that are triggered not by a particular line of code, but by the change in value of a particular variable in code. For example, let's say that you've got a large program and think that your variable iCount is being changed unexpectedly. You can set a regular breakpoint (at any line number), run the debugger to the breakpoint, and then add a watch to iCount.

The debugger will then always break whenever the value of iCount changes.

To set a watchpoint, add a normal breakpoint to your code and run the debugger. On the *Debug* window, click on the **Watch** tab. Then, click anywhere within the **Watch** tab space. An *Add Watch* (shortcut : **F4**) menu should appear.



Once you select *Add Watch*, a dialog box should appear:

Variable to modify:		
New value:		
Break on Write		

Type the name of your variable in the dialog, and press OK. If you select a word in the current source file and press **F4**, it will add a watch of the selected text without asking for a variable name. A breakpoint can be generated when the value of the variable is written, read, or both.

Once a watchpoint variable is set, it will continue to be monitored in future debugging runs. You can also modify the value of your watched variable at any time during debugging by right clicking on the variable name in the *Debug Watch* window and selecting *Modify* value. Note that you are changing this value outside of your source code. The next time you run the debug, the value you entered is lost and the program executes normally.

Important Notes:

- When using pointers to structures or classes, if you want to display all the members of variable *my_pointer* then you need to watch **my_pointer* (*'*'* is the value-operator). Watching only *my_pointer* would just display the address contained in *my_pointer*.
- Sometimes the debugger may not know the type of a pointer, and cannot display all of the members of the pointed structure or class. You can bypass this problem by casting your watched variable. For example, if the debugger cannot show the contents of *my_pointer* of type MyPointer, you could try adding the watch variable: *(*MyPointer* *)*my_pointer*
- If you are watching a wxString variable, you might find that when the string changes, the debugger stops inside one of wxWidgets' string functions and pressing **F7** has no effect. The solution is to set an ordinary breakpoint on the next line and press **F8** to continue to the new breakpoint. *This happens because GDB is a general-purpose debugger and it doesn't know that you are not interested in the internal workings of wxWidgets. It is normally seen as undesirable behaviour but it is not a fault.*

Backtracing

Backtracing is the debugging concept that tells you which functions were called before reaching a breakpoint or an interruption (like an access violation or other runtime error).

Let's do a simple test. Move the <u>breakpoint</u> in your sampleDebug project to the printf line within the test2 (line 12). Then, re-run the debugger (shortcut: **F8**). If you look at the *Debug* window in the *Backtrace* tab, you should see something like this:

Debug			X
廦 Compiler 📶 Compile Log 🛛 🔯 Debug 🔎 Find Results 🍫 To	p-Do List		
Backtrace Local Variables Threads Watches Output			
Function Arguments	File	Line	
test main	main.cpp main.cpp main.cpp	12 19 33	

This correctly shows the list of functions that have been called (since the start of the program) before reaching the breakpoint. Clicking on a function in this list will bring you to its implementation in your source code. So, for example, if you click on test, the IDE will take you to line 19 in the source code.

Now uncomment lines 24-27 in the function test. When you try to debug this time, the program will eventually cause an access violation and end. The IDE should display which line triggered the error and the backtrace should display the functions called to reach it.

Using CPU window

wxDev-C++ provides a CPU window to expert developers who want access to the status of CPU registers, memory and instructions.

To show the CPU window, wait for a breakpoint or interruption to raise in your program and go to the Debug menu, then click View CPU Window. A window similar to this will appear, but the memory pane will be blank:

🔡 СРИ													
Disassembly	[1		_		Registers	Refresh
Source File:	main.cpp								Refre	sh		Register	Value
File: main.cpp										~		eax	0x0000000f
Line: 9 Eurotion: 75test2Pi												ecx	0x77c418bf
0x004013ce	O push	v %ebp										edx	0x77c61b78
0x004013cf	1 mov	%esp,%ebp										ebx	0x7ffde000
Line: 10	3 SUD	\$UX8,%esp										esp	0x0023ff10
0x004013d4	6 movb	5 \$0x72,-0x1(%eb	p)									ebp	0v0023628
Line: 11	10 mov	NuQ(Vabo) Vadu								=		esi	0-00610072
0x004013db	13 mov	Oxc(%ebp),%eax										esi	0x00010073
0x004013de	16 mov	%eax,(%edx)										eai	0x005c0057
Line: 12 0v004013e0	18 movi	\$0v444000 (%ext	5									eip	0x00401407
0x004013e7	25 call	0x416530 < printf>	J)									eflags	0x00000296
Line: 13												CS	0x0000001b
0x004013ec	30 leave 31 ret	3										SS	0x00000023
0x00401360	51 16(~		ds	0x00000023
Momoru					_		3	1			8	es	0x00000023
Address:	&chTest2			Count:	64				Refre	sh		fs	0х0000035
0x0023FF00		72 2	28 ff	23 00	07	14	40	00	r(#	0		gs	0x00000000
0x0023FF10 24	ff 23 00 0	5 00 00 00 2	28 ff	23 00	e0	ff	23	00	\$# (#	#			
0x0023FF20 94	5c c3 77 0	5 00 00 00 3	58 ff	23 00	53	14	40	00	\w X#S	0			
0x0023FF30 42	4e c3 77 a	0 cf 40 00 s	a0 cf	40 00	41	14	40	00	BN w @ @ A	0			
0x0023FF40 a0	CI 40 00 S	00 30							1 w 9				

On the top left, you can find the assembler instructions of the current function. You can display the assembler code of any other function by typing its name in the Function field, then pressing the Enter key.

On the bottom left is the Memory pane, which you can use to display the contents of memory. You can view any *accessible** area of memory. For example, say we want to see where the character chTest2 is stored. (You will

need to step through the program and stop at line 12 to see this example). Type "&chTest2" into the address box and click "Refresh". The number '72' and the character 'r' will appear. This is the value of chTest2 in hexadecimal and as a character. Change *count* to 64 and refresh. Now you will see chTest2 and the following 63 bytes displayed, in hexadecimal in the left-hand block and the corresponding characters in the right-hand block. You can obtain the address where chTest2 was stored by reading off the address and counting columns to the first value (GDB converted what you typed in the address box to the numeric address, you are likely to see a different address to that shown here). You can directly type addresses in decimal ('2359047') or hexadecimal ('0x23FF07') format as well as indirectly ('&chTest2').

* Some memory areas are inaccessible and cannot be displayed.

On the right, are the contents of the CPU registers.

Debugging multi-threaded programs

The GDB Debugger manual describes multi-threaded programs:

" ...a single program may have more than one thread of execution. Th of threads differ from one operating system to another, but in gener single program are akin to multiple processes — except that they sha (that is, they can all examine and modify the same variables). On th thread has its own registers and execution stack, and perhaps privat

Debugging a multi-threaded program is broadly similar to debugging a singlethreaded program, however a number of points must be borne in mind:

- A breakpoint may be set in any part of the program in the normal way. If the breakpoint is set in a function or method that was or will be started as a separate thread, then execution halts as normal when the breakpoint is hit. If the function or method executes in several threads, then execution halts each time the breakpoint is hit, no matter which thread it is in.
- When a thread stops, usually because a breakpoint or watchpoint is hit, then all currently executing threads stop. Only the thread that initiated the stop is guaranteed to stop at a source line. The other threads may stop anywhere, including part-way through a statement.
- The thread that initiated the stop becomes the currently active thread and the focus of debugging.
- The currently active thread is marked with an asterisk '*' in the Threads tab.
- When a thread is started with "Debug" [F8], "Next Step" [F7], "Step Into" [Shift+F7] or *any* other command that causes execution to begin, <u>ALL</u> stopped threads will re-start.
- *A single step might not complete*. It is possible for a breakpoint or another cause in another thread to halt execution before the first thread completes the step you requested.
- It is not possible to separately identify watched variables that have the same name, but which are in different threads; hence the program will stop when any instance of the watched variable changes. However, the thread in which the watched variable changed will become the current thread, and both the Local Variables tab and the Watches tab will show the values of the variables in that current thread.

• Unless you examine memory directly, the CPU window also operates on the current thread. Unlike Local Variables, Watches and Threads, the panes of the CPU window do not refresh automatically and you must click Refresh on each pane as appropriate.

Integrated Templates

wxDev-C++ uses a template system to allow the user to more easily create specific projects. By this point you've encountered templates without knowing it. When you select *File*, *New*, *Project* on the IDE menu, the project type options listed in the dialog box (*e.g.* Console application, wxWidgets frame, etc.) are preset templates.

ew project						X
Basic Intro	duction Mul	ltiMedia				
Empty wxWidge Description:	wxWidgets Dialog	wxWidgets Frame	Console Application	Static Library	dir DLL	
Project optio Name: Project1	ns:			<u>C</u> Project <u>M</u> ake Default La	⊙ C <u>+</u> + Projo anguage	ect
			<u></u>	k Car	ncel	Help

Templates usually include two components:

- 1. A set of source files These source files contains the minimal amount of code needed to build the application
- 2. Project file A preset project file (extension *.dev*) containing all of the compiler and linker options necessary to build the application.

Templates are usually stored within the Templates subdirectory under the wxDev-C++ installation directory. Let's look at a simple example of a template from that directory.


[Template] ver=3 Name=Console Application IconIndex=1 Description=A console application (MSDOS window) Catagory=Basic [Project] UnitCount=1 Name=Console App ProfilesCount=2 ProfileIndex=0 [Unit0] CName=main.c CppName=main.cpp C=consoleapp_c.txt Cpp=consoleapp_cpp.txt [Profile0] ProfileName=Mingw 3.4.2 Type=1 ObjFiles= Includes= Libs= PrivateResource= ResourceIncludes= MakeIncludes= Compiler= CppCompiler= Linker= PreprocDefines= Icon= ExeOutput=Output\MingW ObjectOutput=Objects\MingW OverrideOutput=0 OverrideOutputName= HostApplication= CommandLine= UseCustomMakefile=0 CustomMakefile= IncludeVersionInfo=0 SupportXPThemes=0 CompilerSet=0 compilerType=0

[Profile1] ProfileName=MS VC++ 2005 Type=1 ObjFiles= Includes= Libs= PrivateResource= ResourceIncludes= MakeIncludes= Compiler= CppCompiler= Linker= PreprocDefines= Icon= ExeOutput=Output\MSVC++2005 ObjectOutput=Objects\MSVC++2005 OverrideOutput=0 OverrideOutputName= HostApplication= CommandLine= UseCustomMakefile=0 CustomMakefile= IncludeVersionInfo=0 SupportXPThemes=0 CompilerSet=1 compilerType=1

The template above sets up a generic console project (i.e. a DOS shell application). In the [Project] section, one source code unit and two compiler profiles are defined. The source code unit is further specified as consisting of a C++ skeleton code which is derived from the file consoleapp_cpp.txt and renamed as file main.cpp. The two profiles are further specified as a MingW gcc and a MS VC++ 2005 compiler profile.

Using a text editor, you can easily create your own templates for use with wxDev-C++. However, wxDev-C++ can also automatically create templates for you based on your existing projects. For example, let's say that you frequently like to use the Boost, wxWidgets, and SQLite libraries in your projects. You'd like to create a Boost/wx/Lite (let's call it "BWL") template that sets up all of the includes, switches, and base code so that whenever you create a new project, you

can have the template handle the basic setup and jump right into the main programming. Here's how to create your BWL template:

- 1. Create the most basic project you can that correctly compiles, links, and executes using the Boost, wxWidgets, and SQLite libraries.
- 2. On the *File* menu select *New* and then *Template*. A new template window

Name:	Custom proje	ect1 -
Description:	This is a custo	om project.
Category:		•
New project nar	me: Custom proje	ect
Icons:	late icon	Library
EmptyNewr	project icon	En Browse
	nojecticon	1 Remove

- 3. In the **Template Info** tab, you can specify the name, description, type, and icons associated with this template.
- 4. In the **Files** tab, you can select which files of the current project you want to include in the template. Whenever you create new projects with this template, those files (and their contents) will be generated in the new project.
- 5. In the **Extras** tab, you can specify extra compiler and linker switches/parameters that you wish to include in the template for new projects.

Once you've filled out all of the necessary information for your new template, hit the **Create** button and save the template with a unique name (e.g. *BWL-template.dev*).

To test your new template, close the current project and try to create a new project (*File*, *New*, *Project*). Your BWL template should appear as an option in

the list and can be used like any other.

Getting and Installing Packages

wxDev-C++ makes use of Dev-C++'s <u>DevPak</u> system for updating and improving the program. For example, if you need to use SQLite or OpenSSL in your programs, there are probably DevPaks out there that will allow you to automatically add the header files, libraries, and templates needed to easily add these functions to your programs. DevPaks are also used to automatically update the <u>IDE</u> or the <u>RAD</u>.

What's in a DevPak?

A DevPak is really just a compressed package containing files that will be copied to your computer. The DevPak creator first creates a file (extension *.devpackage*) that contains what files to include in the DevPak and where to copy them on the target computer. Those files are then added to a single file using the <u>tar</u> archiving tool (extension *.tar*) and that single archive file is compressed using the <u>bzip2</u> tool (extension *.tar.bz2*). The single, compressed archive file is given the extension *.devpak*, but it is actually just a tar'd, bzip2'd archive similar to other packages you may have downloaded. Although you can create DevPaks manually, the developers of Dev-C++ made the process even easier by providing a <u>GUI</u> for making DevPaks called PackMaker. You tell PackMaker what files to include in your DevPak and in what directories to install those files, and PackMaker handles the archiving for you.

Where do I get DevPaks?

The easiest way to get DevPaks is to use wxDev-C++'s web update feature. WebUpdate searches a list of DevPak servers and tells you if the ones online are newer than the ones you already have installed. To activate WebUpdate, go to the *Tools* menu and select *Check for Updates/Packages*.

vxDev-C++ DevPak server			•	Check for U	pdate
roups: Selecti All groups> 🗸	on: 0 files selecte Disconnecte	ed, 0 KB (0 Byte d	s)		
vailable updates list:	Version	Installed	File cize	Date	
bringtroom	1.2	Instancu	67 VR	2008-02-07 12:10	_
	20	7.0	1802 KB	2008-03-09 10:30	1
wwWidgets MS VC 2005	2.8.9	1.0	8206 KB	2008-09-27 15:00	
wxWidgets Contribs MS VC 20	05 2.8.9		1725 KB	2008-09-27 15:00	
wxWidgets Extras MS VC 2005	2.8.9		2536 KB	2008-09-27 15:00	
- ·			1000 100		
le description:					
in stream for wwWidgets (wwbz	instream), Works	iust like wxtarst	tream except t	hat it uses Bzip	

You can select the online DevPak server in the dropdown box and then click the **Check for Updates** button to download a list of available DevPaks on that server. Note that we suggest using DevPaks from the wxDev-C++ server because we know they'll work with the wxDev-C++ IDE. Other DevPaks may require some manual intervention to get the correct paths for the compiler and linker.

Once you find a DevPak you want to download, click on the empty checkbox to the right of its name and then click the **Download Selected** button at the bottom, right of the window. The DevPak will be download from the webserver and will begin the install process:

Installation V	Nizard	×
	ev-C++ Package Installation Wizard bzipstream installation procedure.	
	Welcome to the Dev-C++ Package Installation Wizard. This Wizard will install wxbzipstream 1.2. Please press Next to continue with the installation, or press Cancel to stop the installation now.	þ
	wx stream with Bzip compression	
		-
> Welcome Readme License Installing Finished		
About	< <u>Previous</u> <u>Next ></u> <u>Cancel</u>	

In the above case, I've selected the wxBzipStream DevPak to install. There's usually a **Welcome** screen that explains what the package will install and screens for the **ReadMe** and **License** files that typically come with packages. Continue the DevPak installation by clicking on the **Next** button until you are told that the DevPak has installed successfully. Once finished, our wxBzipStream library should be installed on our computer and ready for use in our programming.

Managing DevPaks

- How do I know that my new DevPak really installed correctly?
- What happens if I want to uninstall it?

To manage your DevPaks, the Dev-C++ developers created a DevPak manager called PackMan. To activate PackMan click on the *Tools* menu and select *Package Manager*.

Dev-C++ Package Manage	er (version 2.2.6)	A Company	A Complete A				x
Package View Help							
Install Verify Remove	🗾 င်္ဂလ် လ်လ် Help About	in the second se					
Package Details <							
wxbzipstream	binutils	wxbzipstream	Dev-C++ Help file	Dev-C++ examples	gcc-core	gcc-g++	
1.2						DEV	
wx stream with Bzip	GNU Debugger	Dev-C++ Language	GNU Make	mingw-runtime	Programming with wxDev-C++	Dev-C++ Templates	
		DEV					
	win32api	wxDev-C++ release candidate	wxWidgets common	wxWidgets Contribs common	wxWidgets Extras common	wxWidgets Mingw gcc	
	wxWidgets Contribs MinG	wxWidgets Extras MinGW gcc	wxWidgets Samples	wxWidgets MS VC 2008	wxWidgets Contribs MS	wxWidgets Extras MS VC 2008	
Ready.			10				.đ

As you can see, the wxBzipStream package is shown along with the other packages installed on my system. In the notebook page to the left of the screen, you'll see that the **General** tab lists the name, version, and description of the DevPak. If you click on the **Files** tab, you'll see a list of the files that are associated with this DevPak and where they have been installed on your computer.

Verify To verify that the DevPak successfully installed all of its files to your computer, select the DevPak's icon and click on the **Verify** tool in the toolbar. A message box should appear telling you if any files that should be in the DevPak don't appear on your computer in their correct location.

8

Remove To uninstall the DevPak, select the DevPak's icon and click on the **Remove** tool in the toolbar. PackMan will delete all of the files from your computer that were a part of this DevPak.

Compiler Options

As the name implies, the Compiler Options dialog allows you to configure <u>global</u> settings for the compilers you will use to build your programs.

To access the dialog, click on the *Tools* menu, then select *Compiler Options*.

Compiler Options	×
Compiler Set: Default GCC compiler 💌 🛨	×I
Compiler Settings Directories Programs wxWidgets	
Compiler set to configure:	
Compiler Type: MingW	· 🗟
Use fast but imperfect dependency generation	
Compiler Command Line Add the following commands when calling compiler:	
	*
	-
Add these commands to the linker command line	
	*
	Ŧ
Add these commands to the make command line:	
	^
	-
Compile Delay 0	
Default	Help

Compiler Options

Compiler tab

Compiler Set

The first tab in the Compiler Options dialog allows you to specify the name of your compiler set and any additional commands you wish to add to the compiler or makefile instructions. Recall that wxDev-C++ allows you to use different compiler sets to build the same project.

Note that the name of the compiler set is just a tag. It does not have to be the name of the actual compiler you wish to use. For example, instead of the set name "Default GCC compiler", I can define a compiler set named "Best compiler in the World 1", but use the compiler type of MingW (i.e. the MingW gcc compiler). The buttons 4×10^{10} allow you to add, delete, and rename a compiler set respectively.

Remember that this configuration window will change the <u>global</u> settings used by all of your projects with this compiler set. If you want to add custom compiler/linker settings for a particular project and not affect other projects, then you want to go to the *Project* menu and select <u>*Project Options*</u>.

Compiler Options
Compiler Set: Default GCC compiler
Compiler Settings Directories Programs wxWidgets
Compiler set to configure:
Compiler Type: MingW
Use fast but imperfect dependency generation
Compiler Command Line Add the following commands when calling compiler:
Add these commands to the linker command line
A
· · · · · · · · · · · · · · · · · · ·
Add these commands to the make command line:
· · · · · · · · · · · · · · · · · · ·
· · · · · · · · · · · · · · · · · · ·
Compile Delay 0
← <u>D</u> efault <u>✓ Ok</u> <u>X</u> <u>C</u> ancel <u>? H</u> elp

Compiler set to configure

Here, you actually specify which type of compiler you wish to use with this compiler set. The default compiler that comes bundled with wxDev-C++ is MingW gcc. We also support MS VC 2003, 2005, and 2008 compilers.

Add the following commands when calling compiler

Here you can specify additional command-line options to be passed to the gcc compiler when compiling your project or file. Check the <u>gcc manual</u> for a list of command-line arguments.

Add these commands to the linker command-line

Here you can specify options to be passed onto the linker (ld.exe).

Compile delay

This option is present to provide a delay before compiling. Normally, you will not use this. If *make* complains of the timestamp being invalid, try specifying a delay here.

Use fast but imperfect dependency generation

By default, Dev-C++ will check all files and headers for dependancy information, and update the makefile accordingly. If you find that it's taking too much time, you can prevent this by enabling this option.

Compiler Options

Settings tab

The compiler settings will, of course, depend on which compiler you are using (in most cases Mingw gcc or MS VC). I'm only going to explain the MingW gcc settings in this help file. However, Microsoft provides a very informative <u>online</u> <u>library</u> if you need help with MS VC compiler settings.

ompiler Options		X
Compiler Set: Default	GCC compiler 🗨	+ 🗙 🗉
Compiler Settings	Directories Programs wxWidgets	
C compiler: C++ compiler: Code Generation Code profiling Linker: Optimization	Support all ANSI standard C programs Attempt to support some aspects of traditi Inhibit all warning messages Displays one error per line	No No No

NOTE: A more in depth explanation of these compiler options can be found in the <u>gcc manual</u>.

C Compiler

Support all ANSI standard C programs

Option -ansi: Will attempt to be as ANSI conformant as possible.

Attempt to support some aspects of traditional C pre-processors

Option -traditional-cpp : Will attempt to make the pre-processor behave as traditional ones do.

Inhibit all warning messages

Option -w: No warnings will be displayed.

Displays one error per line

Option -fmessage-length=0 : Displays one error per line!

C++ compiler

Turn off all access checking

Option -fno-access-control

Accept \$ in identifiers

Option -fdollar-in-identifiers: Will allow \$ to be used in variable and function names. Cannot be used with GCC 3.2.

Code generation

Enable exception handling

Option -fexceptions

Use same size for double and float

Option -fshort-double

Put extra commentary information in the generated assembler

Option -fverbose-asm: The intermediary assembler files generated will have extra comments. Only useful if they're being saved (using the -S command-line option).

Code profiling

Generate profiling info for analysis

Option -pg: Writes extra information into the generated program files to use with the profiler. The profiler lets you see where maximum amount of the program's execution time is being spent, so that when you're optimizing the program, you optimize only the parts that make the difference. This option should be disabled when building retail/final versions. It should be used only in debug builds.

Linker

Link an **Objective C program**

Option -lobjc

Generate debugging information

Option -g3: Writes debug information into the generate binaries. This lets you debug it with the <u>integrated debugger</u>.

Do not use standard system startup files or libraries

Option -nostdlib: Do not use the standard system startup files or libraries when linking. No startup files and only the libraries you specify will be passed to the

linker. The compiler may generate calls to memcmp, memset, memcpy and memmove. These entries are usually resolved by entries in libc. These entry points should be supplied through some other mechanism when this option is specified.

Do not create a console window

Option -mwindows: Instructs gcc to build a Windows GUI application without a console window.

Strip executable

Option -s: Remove all symbol table and relocation information from the executable.

Optimization

Optimization can make your code execute faster, but requires more time to compile.

Perform a number of minor optimizations

Option -fexpensive-optimizations: Perform a number of minor optimizations that are relatively expensive. Enabled at levels -O2, -O3, -Os.

Further Optimizations

Here, you can select the optimization level.

Optimize

Option -O1: With -O1 (or just -O), the compiler tries to reduce code size and execution time, without performing any optimizations that take a great deal of compilation time.

Optimize more

Option -O2: GCC performs nearly all supported optimizations that do not involve a space-speed tradeoff. The compiler does not perform loop unrolling or function inlining when you specify -O2. As compared to -O, this option increases both compilation time and the performance of the generated code.

Best optimization

Option -O3: Optimize yet more. -O3 turns on all optimizations specified by -O2 and also turns on the -finline-functions, -funswitch-loops, -fpredictive-commoning, -fgcse-after-reload and -ftree-vectorize options.

Compiler Options

Directories tab

These are the directories the IDE will search when it is building your program.



Binaries

Specifies the locations of the compiler, linker, and other executables.

Libraries

Specifies the locations of the library files (*.a, *.dll, *.lib, *.def).

C includes

Specifies the locations of the headers (*.h) for C programs.

C++ includes

Specifies the locations of the headers (*.h, *.hpp) for C++ programs.

Resource includes

Specifies the location for the resource files (*.rc) for Windows programs.

Compiler Options

Programs tab

Compiler Options		X
Compiler Set: Defau	ılt GCC compiler 🗾 🚽	×I
Compiler Settings	Directories Programs wxWidgets	
You may want to c Dev-C++ (for exam	hange the programs filenames that are used in ple when using a cross compiler) :	
C compiler:	gcc.exe	
C++ Compiler:	g++.exe	
Make:	mingw32-make.exe	
Resource Compiler:	windres.exe	
Linker:	dllwrap.exe	
Debugger:	gdb.exe	
Code Profiler:	gprof.exe	
← <u>D</u> efault	Ok X Cancel	Help

Here you can specify the filenames of the different compiler executable components. Note that wxDev-C++ uses the MingW make system for all of its builds.

Compiler Options

wxWidgets tab

This is where you can store the information about the wxWidgets libraries you want to use. The information here will be used to change the values passed to the linker in the MingW makefile. For the current example, the program would be linked to to the library libwxmsw28.a for gcc and wxmsw28.lib for MS VC. If you update your wxWidgets libraries (or want to use your own libraries), then you'll need to update this information in order to pass the correct library to the makefile.

Compiler Set: Defaul Compiler Settings	t GCC compiler Directories Programs	wxWidgets	▼ + × □
Minor: 8 Release: 2			
Features Unicode Suppo Monolithic Libr Debug Build	rt ary		
Library Type Static Import Li Dynamic Librar	ibrary y (DLL)		
- Default		Can	al 7 Help

Environment Options

General options tab

To access the Environment Options dialog, click on the *Tools* menu and select Environment *Options*.

External programs	File Associati	ons CVS Support
General	Interface	Files & Directories
 Allow only one insta Create Backup Files Minimize on Run 	nce of wxDev-C++	
Show Toolbars in Fu	II Screen	
🗖 Show Menu in Full S	Green Mode	
Double Click to Ope	n Project Manager F	iles
Auto Open C All Project Files C Only First Project F None	ile	ug variables browser /atch variable under mouse eport watch errors

Allow only one instance of wxDev-C++

Allows only one instance of the IDE to be running.

Create Backup files

If enabled, whenever you save a source file inwx Dev-C++, a backup copy will be saved along with it. This backup will be overwritten on successive saves.

Minimize on run

If enabled, wxDev-C++ will minimize itself when you execute your program from within it (using the `Run' command under `Execute').

Show toolbars in Full-screen

By default, toolbars are hidden when wxDev-C++ is made full-screen. If this option is enabled, they will be shown all the time.

Double-click to open project-manager files

If this option is enabled, you'll need to double-click on the nodes in projectmanager to open them in the editor. Otherwise, you would single-click.

Auto-open...

Here, you can choose what files are automatically opened in the editor when you open a project.

Debug variables browser

If "Watch variables under mouse" is checked, variables under the current mouse position will be created as watch variables during a debugging session.

Environment Options

Interface

External programs	File Asso	ociations	CVS Sup	port
General Ir	nterface		Files & Directori	es
Max Files in ReOpen Menu	1	0		
Message Window Tabs:	Ī	ор	•	
Language:		nglish (O	riginal) 💌	[
Theme:	1	lew Look	•	ĺ
No splash screen on star Use XP Theme	tup			
✓ Use Native Docking Win	dows			
Open/Save Dialog Style Windows 2k (sidebar) Windows 9x Windows 3.1		Compilati <u>Show</u> <u>Auto</u>	on Progress Wind during compilatio :lose after compil	low – on le

Max files in re-open menu

Here, you can specify the number of files wxDev-C++ keeps track of in the Reopen menu. Older files are forgotten first.

Message Window Tabs

Sets the position of the message window tabs (Compiler, Resources, Compile Log, Debug, Find, ToDo List)

Language

Select your language here.

Theme

Select the theme you want wxDev-C++ to use.

No splash screen on startup

If enabled, it prevents wxDev-C++ for displaying a logo when it starts up.

Use XP theme (WinXP only)

If enabled, it makes wxDev-C++ use Windows XP themes (bitmap title-bars, buttons, etc.).

Open/Save dialog style

Here you can select the type of file open/save dialog you want to see.

Compilation progress window

Allows you to select whether you wish to see the compiler output log at compile time and if you wish it to auto-close.
Environment Options tab

Files and Directories

This is where you can modify the default directories that the IDE uses to store configuration files, templates, and user data.

External programs	File Associatio	ons CVS	Support
General	Interface	Files & Dire	ectories
 Alternate Configuration 	on File		
Use this alternate of	configuration file		
			<u></u> ;;8]
User's Default Directory			
Templates Directory			
Templates\			
			_>=
Icon Library Path			
licons			
Language Files Path			
Lang\			
Splash Screen Image			

Alternate configuration file

wxDev-C++ typically saves your preferences to a settings file (*devcpp.cfg*) located in the user **AppData** directory. You can change where the IDE looks for this settings file. This is useful if you want to have all of the IDE files in one directory for portability. (Some users have been able to use wxDev-C++ on a thumbdrive that they move from computer to computer without having to install it on the computer).

User's default directory

Specifies the default location that the IDE uses (when creating projects, opening, saving, etc.)

Templates

Specifies the location of the Dev-C++ project templates (the ones displayed when you click on `New Project').

Icon library path

Specifies the location of the icons for use in your projects.

Language Files path

Specifies the location of the Dev-C++ language files.

Splash screen image

You can specify an alternate splash-screen bitmap here.

Environment Options

File Associations tab

Here, you can select one or more file types which the IDE will associate itself with, so that the next time you open an associated file in Windows Explorer, wxDev-C++ will be opened up automatically and load that file.

General	Interface	File	s & Directories
External programs	File Associa	tions	CVS Support
e Types			
C Source File (*.c	-)		
C++ Source File	(*.cpp)		
C Header File (*.	h)		
C++ Header File	(*.hpp)		
Dev-C++ Project	File (*.dev)		
Resource Source	File (*.rc)		
Dev-C++ Templa	ate File (".template)		
lust check or un-che	ck for which file type	⊳s Dev-C+	+ will be
ust check or un-che	ck for which file type	es Dev-C+	+ will be
ust check or un-che registered as the defa	ck for which file type ault application to op	es Dev-C+ ben them.	+ will be
ust check or un-che egistered as the defa	ck for which file type ault application to op	es Dev-C+ ben them.	+ will be

Environment Options

CVS Support tab

Concurrent Versioning System (CVS) is a way to store your project files and keep track of any changes you (and other developers) make. It allows you to not only backup your current files, but also to rollback to earlier versions of your files and undo changes.



CVS program file

Enter the name and directory of the CVS program executable.

Compression level

Specify the compression level to be used.

Use <u>SSH</u> instead of RSH

Use the SSH program to connect

Editor options allow you to configure how you wish your source code to display within the IDE. wxDev-C++ allows you to configure code formatting, line numbers, code completion, syntax highlighting, and many other options that are useful when coding large programs.

To access the Editor Options go to the *Tools* menu and select *Editor Options*.

Auto Indent			
		Enha	nced Home Key
Insert Mode		Curse	or Past EOF
Use Tab Chara	cter	Curse	or Past EOL
Smart Tabs		Double Click Line	
Keep Trailing Spaces		Find Text at Cursor	
Backspace Unindents		Scrol	llbars as needed
🔽 Group Undo		☐ Half	Page Scroll
Insert Dropped Files		Scrol	ll Hint
Show hidden I	ine characters	Show	v editor hints
Ensure that file	ends with newli	ne	
 Highlight curr 	ent line		-
Caret			Right Margin
nsert Caret	Vertical Line	-	Visible
Overwrite Caret	Vertical Line	-	Width Color
Highlight mat	ching braces/par	enthes	80 🜲 🔍 🔻
			,

General tab

Editor Options:	Syntax Code		
Auto Indent		Enhanced Home Key	
✓ Insert Mode		Cursor Past EOF	
Use Tab Chara	cter	Cursor Past EOL	
Smart Tabs		Double Click Line	
Keep Trailing Spaces		Find Text at Cursor	
Backspace Unindents		Scrollbars as needed	
Group Undo		Half Page Scroll	
Insert Dropped Files		Scroll Hint	
Show hidden line characters		Show editor hints	
Ensure that file	e ends with newli	ine	
Highlight curr	ent line	—	
Caret		Right Margin	51
Insert Caret	Vertical Line	Visible	
Overwrite Caret	Vertical Line	Width Color	
UVELWITE COLE	wertical Line	80 🛨	
	recipient ar ar ar ar ar	renunes j	
Highlight mat	ching braces/ par		_
Highlight mat	ching braces/par		

Editor options

- Auto Indent Automatically indents a new line according to the current line's syntax
- Insert Mode Text will insert at the current cursor position (not overwrite)
- Use Tab Character Inserts tab character, otherwise inserts spaces
- Smart Tabs When [Enter] is pressed, indents the new line with the same combination of tabs and spaces as the previous line, otherwise indents the new line with the most economical combination of tabs and spaces, depending on the setting of Use Tab Character

- Keep Trailing Spaces
- Backspace Unindents If a tab stands before the current cursor position, the backspace key will remove it
- Group Undo When undoing/redoing actions, handles all successive changes of the same kind in one operation, otherwise handles one operation at a time
- Insert Dropped Files
- Show hidden line characters Displays hidden characters in the editor
- Ensure that file ends with a newline
- Highlight current line Highlights the line at the current cursor position
- Enhanced Home Key
- Cursor Past EOF Allows you to place the cursor past the end-of-file character
- Cursor Past EOL Allows you to place the cursor past the end-of-line character
- Double Click Line
- Find Text at Cursor If checked, the "Find" dialog will use the text at the current cursor position as the search term.
- Scrollbars as needed If checked, scrollbars will only be displayed if there is text beyond what's visible in the window.
- Half Page Scroll
- Scroll hint
- Show editor hints

Caret

- Insert Caret If text mode is set to insert, then use this caret graphic to mark the current position.
- Overwrite Caret If text mode is not set to insert (i.e. typing will overwrite text), then use this caret graphic to mark the current position.
- Highlight matching braces/parenthesis Highlight the matching braces in a block of code

Right Margin

- Visible If checked, a vertical line is displayed to denote the right margin in the editor
- Width Sets the line width marker

• Color - Color of the right margin marker

Display tab

Font Courier New	Size Tab Size: \checkmark 10 \checkmark 4 \checkmark
	Editor Font
Gutter	
✓ Visible	Line Numbers
Auto Size	Start at Zero
 Use Custom Font Draw with Gradient 	Show Leading Zeros
Font	Size Width
Terminal	▼ 9 ▼ 32 ↓
	Gutter Font

Editor font

As the name implies, this allows you to select which font you wish to use for the editor. It also also you to specify how many spaces are used for tabs.

Gutter

- Visible The gutter is the left margin of the text editor. If this is checked, the gutter will be displayed.
- Autosize The width of the gutter will change to best fit the text.

- Use custom font Allows you to specify a gutter font that is different than the text editor
- Line numbers If checked, it will display the line numbers of the text
- Start at zero Line numbers typically start at 1. If this box is checked, they will start at 0.
- Show leading Zeros Left pads 0's to the line numbers (*e.g.* 0020) to make all line numbers the same width.

Syntax tab



Syntax highlighting allows you to format your display of C/C++ source code using different fonts, colors, and backgrounds. Note that the wxDev-C++/Dev-C++ editor only knows C/C++ syntax highlighting and probably won't work for other programming languages. The **Type** listbox allows you to configure each programming language class (comments, assembler code, numbers, reserved words, etc.) The **Enabled file extensions** text box specifies which file extensions apply to these definitions. The highlighted code at the bottom of the tab shows a preview of how C/C++ source code will be displayed with the current settings.

Code tab

This allows the user to define a menu option that adds code snippets while you are programming. The idea is that if you have code that you frequently use (e.g. a class or function that you almost always add to your programs), then you can set up a menu selection that automatically adds this code to your current project at the current cursor position.

Jeeum Description	Add
	Tennove

To add a code snippet to the menu, go to the *Tools* menu and select *Editor Options*. Then, click on the **Add** button. The *Add Code Insert Entry* dialog will allow you to name the new menu entry and a description of what the entry will do. Then, add the code snippet to the **Code** textbox. When you click the Ok button, your code snippet will be added to the *Edit* menu under *Insert*. Selecting

that menu item will insert your **Code** into the IDE at the current cursor position.

Class browsing tab

Editor Options		*		×
General Disp	olay Syntax Code	Class browsir	ng	1
Class brows	ing Completion			
Engine op	tions ocal files referenced in #	include's		
View optic	lobal files referenced in	#include's		
Use co	lors nherited members			
Sample: [⊡ 🎸 Class1 : class ⊡ 🎸 Class2 : class			
		<u>0</u> k X	<u>C</u> ancel	? Help

The Class Browser is a hierarchical tree that displays all of the classes, functions, global variables, and structures that are used in your project. It's a nice way to graphically display the major players in your programs. The *Class Browser* is displayed in the *Project Inspector* window under the **Classes** tab.

NOTE: Class browsing MUST BE ENABLED to use the wxDev-C++ visual designer.

In the **Engine options** section, you can configure the class browser to scan both local and global include files for their major players. If you chose to scan for global include files, then you will also include in the class browser the classes and functions from the 3rd party libraries you link with your program (e.g. wxWidgets libraries, SQLite libraries, etc.).

In the **View** options section, checking the **Show inherited members** box will cause the class browser to also display any classes from which your classes are dervied. For example, if you create your own wxFrame class called myFrame1, the browser windowwould typically only show the members of that class that you've specified in your project. With the **Show inherited members** checked it will also display the members that were inherited from the generic wxFrame class.

Code completion tab

Editor Options	X
General Display Syntax Code Class browsing Image: Enable class browser Class browsing Completion Image: Class browsing Completion Class browsing Image: Class browsing Solo Image: Class	
Files in cache:	<u>A</u> dd Clear
Qk X Cance	<u>?</u> <u>H</u> elp

Code completion allows you to program C/C++ more efficiently. When enabled, code completion will try to offer suggestions for completing your code as you type it. For example, if you were working on a wxFrame project and typed:

this->

in the source code, a dropdown box will appear offering properties and functions that can complete the command. The **Delay** value in code completion specifies the amount of time (in milliseconds) which a user must pause typing for the code completion to be activated. For example, with a delay of 500, the user will have to pause typing for 500 ms (or half a second) before the code completion

command is initiated; otherwise, the code completion will be skipped.



Semi-auto complete

You can manually invoke code completion with the key sequence **Ctrl+Space**.

Code completion cache

Although code completion is very useful, it takes time to build a list of possible completions to your code. You may be able to speed things up by creating a cache. The cache is loaded when the IDE starts up (so it takes a longer time to start the IDE) so the IDE parses the single cache file instead of the multiple include files. To add include files to the code completion cache, click on the **Add** button and select the file to include in the cache.

 \leq Previous \leq = Home = \geq Next \geq

Configure Shortcuts

You can change the keyboard shortcuts by going to the *Tools* menu and selecting *Configure Shortcuts*.

Configure Shortcuts		
Click on an item and press the shortcut you desire!		
Tip: press "Escape" to clear a shortcut	a	
Menu entry	Shortcut assign	
File Source File	Ctrl+N	
File Project		
File Resource File		
File Template		
File Open Project or File	Ctrl+0	
File Clear History		
File Save	Ctrl+S	
File Save As	Ctrl+F12	
File Save Project as		
File Save All		
File Close	Ctrl+F4	
File Close All		
File Close Project		
File Properties		
File Import MS Visual C++ project		
File to HTML		
File to RTF		
File Project to HTML		
File Print	Ctrl+P	
File Print Setup	T	
	Ok Cancel	

Use the mouse to select the menu entry for which you wish to add or change a shortcut. Then, type the new keyboard shortcut for that menu entry. To delete a shortcut, hit "Esc". Press the "Ok" button when finished.

 \leq Previous \leq = Home = \geq Next \geq

Editor Shortcuts

Actions:

These commands perform various actions.

Command	Description
Control + Shift + B	Go to matching bracket
Control + (number)	Move to marker (number)
Control + Shift + (number)	Set marker (number)
F1	Context sensitive help on word

Editor Shortcuts

Cursor:

These commands control how the cursor behaves.

Command	Description
Left Arrow	Left one character
Right Arrow	Right one character
Up Arrow	Up one line
Down Arrow	Down one line
Control + Left	Left one word
Control + Right	Right one word
Home	Start of line
End	End of line
Page Up	Up one page
Page Down	Down one page
	Left one page
	Right one page
Control + Page Up	Top of page
Control + Page Down	Bottom of page
Control + Home	Abs begin
Control + End	Abs end

 \leq Previous \leq = Home = \geq Next \geq

Editor Shortcuts

Modes:

These commands control modes.

Command	Description
Insert	Set insert mode
	Set overwrite mode
	Toggle insert/overwrite
Control + Shift + N	Selection type is normal
Control + Shift + C	Selection type is column
Control + Shift + L	Selection type is line

 \leq Previous \leq = Home = \geq Next \geq
Editor Shortcuts

Delete:

All the commands having to do with deleting.

Command	Description		
Backspace	Character to left		
Shift + Backspace	Character to left		
Delete	Character to right		
Control + T	Word to right		
Control + Backspace	Word to left		
	From cursor to start of line		
Control + Shift + Y	From cursor to end of line		
Control + Y	Current line		
	Everything in editor		
Enter	Line break at current position, move caret		
Shift + Enter	Line break at current position, move caret		
Control + M	Line break at current position, move caret		
Control + N	Line break at current position, don't move caret		
	Insert character at curent position		
Alt + Backspace	Perform undo if available		
Control + Z	Perform undo if available		
Alt + Shift +	Perform redo if available		

Back		
Control + Shift + Z	Perform redo if available	
Shift + Delete	Remove selection place on clipboard	
Control + X	Remove selection place on clipboard	
Shift + Insert	Move clipboard contents to current position	
Control + V	Move clipboard contents to current position	
Control + Shift + I	Move selection to right	
Control + Shift + U	Move selection to left	
Tab	Tab key	
Shift + Tab	Tab to left	

Editor Shortcuts

Scrolling:

These commands control everything to do with scrolling.

Command	Description		
Scroll Up + Control	Go up one line		
Scroll Down + Control	Go down one line		
Scroll Left	Left one character		
Scroll Right	Right one character		

Editor Shortcuts

Selection:

These commands control how the currently highlighted text behaves.

Command	Description		
Shift + Left	Select left		
Shift + Right	Select right		
Shift + Up	Select up		
Shift + Down	Select down		
Control + Shift + Left	Select word left		
Control + Shift + Right	Select word right		
Shift + Home	Select line start		
Shift + End	Select line end		
Shift + Page Up	Select page up		
Shift + Page Down	Select page down		
	Select page left		
	Select page right		
Control + Shift + Page Up	Select page top		
Control + Shift + Page Down	Select page bottom		
Control + Shift + Home	Select editor top		
Control + Shift + End	Select gotoxy		
Control + A	Select All		
Control + Insert	Select All		
Control + C	Copy selection to clipboard		

Control + V	Paste selection to
	clipboard
Control + X	Cut selection to clipboard

Editor Shortcuts

Visual Designer:

These commands only apply to the wxWidgets visual designer and only work if a component (i.e. widget) has been selected.

Command	Description
Ctrl + Left	Move the component one tick to the left
Ctrl + Right	Move the component one tick to the right
Ctrl + Up	Move the component one tick up
Cttrl + Down	Move the component one tick down
Ctrl + Del	Delete the component
Ctrl + X	Cut the component from the designer
Ctrl + C	Copy the component
Ctrl + V	Paste a copy of the component to the form

ToDo List

wxDev-C++ provides a way to keep track of your "ToDo" list from within your program's comments. This makes it easy to keep track of incomplete sections of your source code.

There are two methods for adding items to a ToDo list:

- 1. Place the cursor somewhere in the source, then either a right mouse click and select "Add ToDo Item" or just Shift+Ctrl+T.
- 2. Just embed C/C++ style comments within your code that are prefaced by the identifier "TODO:". For example,

/* TODO : Add ToDo instructions to help tutorial. */

Project Inspector 9 X	Project1App.cpp	[*] Project1App.h	Project1App.rc	Project1Frm.h	Project1Frm.cpp	Project 1	• (
Project Classes	// Autho	r: tony					
Project1Frm : class	// Creat	ed: 8/23	/2009 7:01:5	55 PM			
Project1FrmApp : class	// Description:						
	11	//					
	//						=
							-
	/* TODO	: Add ToDo in	nstructions	to help tut	torial. */		
	#ifndef	PROJECT1FR	App h				
	#define	PROJECT1FR	App h				
	// DONE:	Get milk at	the store.				
	tifdaf	BODI ANDC					
	#IIUEI #pra	gma hdrstop					
	#E	gaa narooop					-
			III			•	
To-Do List							
📑 Compiler 💱 Resources 📶	Compile Log 🖸	Debug 🔎 Find	Results 🍓 To-	Do List			
Priority Description		Filename			User		
0 Add ToDo instruction	ns to help tutorial.	C:\Users\tony\Do	cuments\junk\Pr	oject1App.h			
Get milk at the store.	-	C:\Users\tony\Do	cuments\junk\Pr	oject1App.h			
Filter Current file only	_]						
Don't show items marked as don	•						
64: 2 Modifiec Insert	39 lines in file						

When the project is reloaded, the "ToDo List" will be displayed. Clicking on the check mark to the left of the ToDo will change mark the item as "Done". Double-clicking on the ToDo item will move the editor to that line in the source.

To specify the priority of the ToDo item, add the identifier " (username#priority#):" to the "TODO :". For example,

```
// TODO (tony#7#): Download the wxDev-C++ help file
```

will set the priority level to "7" and the user to "tony" for this item.

Note that the dropdown box labeled "Filter" allows you to view ToDo's that are

within your current open file or within an combination of files in and out of your project.

Form Designer Options

Designer Options:

Some basic settings for the wxWidgets form designer can be set through the *Tools* menu and selecting *Designer Options*.

orm Designer Optio	ns	X
Designer Options	Code Generation Options	
Form Grid Show Form G Snap to grid X-coordinate grid	rid d interval: 8	Behaviour Floating
Guide Hints		
 ✓ Size ✓ Move ✓ Insert 		
	ОК Х	Cancel ? Help

Grid Settings

In the first box, the grid settings can be modified:

- Show Form Grid Toggle grid on/off
- Snap to Grid Components will be aligned to the nearest grid point
- X-, Y-coordinate grid interval The spacing used between grid ticks

Behaviour Settings

In the second box, the visual designer can be set to "floating". When this box is checked, the visual designer window(s) will be undocked from the IDE. Floating makes it easier to manage large windows. For example, in the image below, the two designer forms (*left*) are floating independently of the wxDev-C++ IDE (*right*).



Form Designer Options

Code Generation Options:

The form designer generates wxWidgets C++ code. Some preferences for the code generation can be set through the *Tools* menu and selecting *Designer Options*.

Form Designer Options
Designer Options Code Generation Options
Code Generation Generate XRC Code String Internationalization: wxT Sizer Options Use wxDefaultPosition Use wxDefaultSize
OK Cancel ? Help

- **Generate XRC Code** When checked, the standard wxWidgets C++ code is replaced by wxWidgets XRC code. <u>XRC</u> is wxWidgets XML-based resource system.
- **String Internationalization** When the code generator passes strings to the wxWidgets component properties, it uses this function as a wrapper. Choices are wxT(), _T(), and _().
- **Sizer Options** Allows the sizer code to use the predefines wxDefaultPosition and wxDefaultSize rather than a numerical value.

Frequently Asked Questions (FAQ)

wxDev-C++ FAQ

Please note that we have a robust <u>wxforum</u> where someone has probably asked your question already. We also have several online <u>tutorials</u> that are a great help to the beginner. Finally, SofT has written a book called <u>"Programming with wxDev-C++</u>".

If you have web access, please consult our <u>online FAQ</u>.

• What is wxDev-C++?

wxDev-C++ is an extension of Dev-C++ with a Form Designer for the amazing cross-platform framework <u>wxWidgets</u>. It contains all the features of Dev-C++ alongside a recent wxWidgets distribution along with a form desinger for rapid application development (RAD).

• Where can I get help for wxDev-C++?

You can either see the available <u>tutorials</u> or post your question in the <u>wxForum</u>.

• Which language is wxDev-C++ written in?

wxDev-C++ and Dev-C++ were written with Borland Delphi 6.0 Personal Edition, a free (albeit closed-source) compiler for Windows (yes, the irony). Lots of open-sourced, third party components like the JVCL and extlib are used.

• Where do I obtain the wxDev-C++ source code?

Have a look at our <u>Subversion repository</u>. You can download the source by clicking on the "Download GNU tarball" link.

• Will there be a release for Linux?

You will have to ask this question to the core Dev-C++ developers. Since wxDev-C++ is based on Dev-C++, a Linux distribution will be a possible only if we have a Linux version of Dev-C++. Some effort (by Tony) have been put in to compile wxDev-C++ with Lazarus failed. Nuklear Zelph has managed to get wxDev-C++ running under Wine on the latest version of Ubuntu linux. You may wish to contact Nuklear on the wxforum if you want to find out how to do this.

• Why does wxDev-C++ take a long time to compile, and the binaries produced so large?

Because MingW gcc tries to emit code that will work on many platforms, it tends to be <u>slow at compiling programs</u>. Using the MS VC compiler may speed up compilation and produce (slightly) smaller binaries.

Large binaries are produced because of static linkage with libstdc++ (which is said to be bulky) and the static, monolithic build of wxWidgets that we include in wxDev-C++. Dynamic linkage to wxWidgets can be done if desired (see the <u>tutorial</u> on building your own wxWidgets library), and switching to STLport is advised if you need to reduce the binary size.

Most people assume that large binaries are a sign of waste and inefficiency. This may be true. However, remember that GUI programs are necessarily large due to the intensive use of graphics (even if you don't take advantage of all that the graphics can do). It's unlikely you'll produce a useful GUI for under 100 Kb or even under 1 Mb (even for a "Hello World!" program). So you can build you own wxWidgets library as a DLL that just includes wxTextCtrl and wxButton, BUT it's not going to be a very useful library for your other programs (which may need more components).

Dev-C++ FAQ

NOTE: This part of the FAQ is a holdover from Dev-C++ 4.9.9.2 and may no longer be up-to-date or relevant.

• Why can't I use conio.h functions like clrsrc()?

Because conio.h is not part of the C standard. It is a Borland extension, and works only with Borland compilers (and perhaps some other commercial compilers). Dev-C++ uses GCC, the GNU Compiler Collection, as its compiler. GCC is originally a UNIX compiler, and aims for portability and standards-compliance.

If you really canot live without them, you can use Borland functions this way: Include conio.h to your source, and add the following file to your project : C:\Dev-C++\include\conio.c (where C:\Dev-C++ is where you installed Dev-C++).

Please note that conio support is not complete.

• My console window keeps closing, how do I change that ?

You can do it this way:

```
#include <iostream>
int main(int argc, char *argv[])
{
    printf ("Press ENTER to
continue...\n");
    getchar (); /* wait for input */
    return EXIT_SUCCESS;
}
```

 After linking, i get an error like C:\DEV-C++\LIB\\libmingw32.a(main.o) (.text+0x8e): undefined reference to `WinMain@16' You probably haven't declared any main() or WinMain() function in your program.

• How can i provide a .def file for my DLL ?

Go to the *Project* menu and select *Project Options*, *Parameters* sheet, **Linker** box . Add to the textbox, --def yourfile.def

• I am having strange problems under Windows XP

Try to run Windows Update and make sure that you have the Program Compatibility updates.

• How do I enable Debugging mode ?

Go to the *Project* menu and select *Project Options* and click on the *Compiler* sheet. In the **Linker** section, put "Yes" to 'Generate debugging information'. Do a 'Rebuild All' and you should be able to debug now.

• When I launch Dev-C++ I get the message saying 'WININET.DLL' or 'MSCVRT.DLL' or 'SHFOLDER.DLL' not found

You are missing a Windows DLL (and probably have a broken version of Windows). Please go to the Microsoft Update site and see if you can fix your installation of Windows.

• How to use assembly with Dev-C++ ?

The "GNU as" assembler uses AT&T syntax (not Intel). Here's an example of such a syntax :

```
// 2 global variables
int AdrIO;
static char ValIO;
void MyFunction(.....)
```

```
{
    ___asm("mov _AdrIO, %dx") ; //
loading 16 bits register
    ___asm("mov _ValIO, %al") ; //
loading 8 bits register
    /*
    Don't forget the underscore _
before each global variable names !
    */
    __asm("mov %ax,%dx") ; // AX --> DX
}
```

• I am using Windows 98 and I cannot compile

Some users have report that you need to apply several patches to your system. Here is the list of them, they can be found on Microsoft Windows 98 download site.

- * 47569us.exe labeled as Windows98SE shutdown
- * dcom98.exe see also this page

* DX81eng.exe - latest version of DirectX (this is 11MB, and cannot be uninstalled without reinstalling Windows 98. You might want to try this one last in case the other above didn't work, as it should update many parts of the system). < Previous < = Home =</pre>

Mailing List / Forum

wxDev-C++

If you have wxDev-C++ specific questions, please post them on the <u>wxforum</u>. This is the preferred mailing list. Please remember to search the forum for the answer to your question before posting it. Chances are that someone has already asked (and answered) it. We also have a <u>Frequently Asked Questions</u> page.

If you're sure that you've found a bug, please post it to our <u>Bug Tracker</u> on SourceForge.

Dev-C++

If you have programming or <u>Dev-C++</u> specific questions, you can post it on the forum or on the mailing list.

Bloodshed Software / Dev-C++ Forum : <u>http://bloodshed.net/forum</u>

Dev-C++ Mailing List : <u>http://bloodshed.net/devcpp-ml.html</u>



wxDev-C++ / Dev-C++ Help File

- Introduction
- **GNU General Public License**
- <u>Credits</u>
- <u>Getting Started</u>
- Basic Steps
 - Creating a Project
 - Adding/Removing files
 - Managing Project Options
 - <u>Project options overview</u>
 - Linking libraries with your project
- Compiling and Running
 - Compiling and linking process
 - Executing your program
- Debugging within Dev-C++
 - Introduction to debugging
 - Sample Program to Debug
 - <u>Debugging your program</u>
 - Integrated debugging commands
 - <u>Breakpoints</u>
 - <u>Stepping through your program</u>
 - Watchpoints
 - Backtracing
 - <u>Using the CPU window</u>
 - Debugging multi-threaded programs
- Dev-C++ Templates and Packages
 - <u>Integrated Templates</u>
 - Getting and Installing Packages
- <u>Compiler Options</u>
 - <u>Compiler</u>

- <u>Settings</u>
- <u>Directories</u>
- <u>Programs</u>
- <u>wxWidgets</u>
- Environment Options
 - <u>General</u>
 - <u>Interface</u>
 - Files & Directories
 - File Associations
 - <u>CVS</u>
- Editor Options
 - <u>General</u>
 - <u>Display</u>
 - <u>Syntax</u>
 - <u>Code</u>
 - <u>Class Browser Tab</u>
 - <u>Code completion</u>
- <u>Shortcuts</u>
 - <u>Actions</u>
 - <u>Cursor</u>
 - <u>Modes</u>
 - <u>Delete</u>
 - <u>Scrolling</u>
 - <u>Selection</u>
 - <u>Visual Designer</u>
- <u>ToDo List</u>
- Visual Designer (RAD)
 - <u>Designer Options</u>
 - Code Generation Options
- Getting help
 - FAQ (Frequently Asked Questions)
 - Mailing List / Forum