# CrossKylix & CrossFPC

Michaël Van Canneyt

31st March 2005

**Abstract**

Borland's kylix product seems to have been abandoned. With it could disappear a nice Delphi add on: CrossKylix. But the future may not be so dark, because a successor is being worked on, ready to replace CrossKylix: CrossFPC.

## 1 Introduction

The future is uncertain for Delphi programmers who welcomed Kylix as a way to port their favourite applications to Linux: Kylix seems to have been abandoned by Borland. At least, no new versions have been announced, Kylix was removed from the latest Delphi release, and no official fixes have been provided to make sure that Kylix-Compiled applications run on the latest Linux kernels - which have a thoroughly changed way of handling e.g. threads.

The more is the pity, since there is a nice (free) product out there which makes it easy to develop Delphi programs for linux, even without leaving the windows environment: CrossKylix is a plugin which allows Delphi developers to cross-compile their application for linux, from within Delphi.

The uncertain future of Kylix made the creator of Cross Kylix (Simon Kissel) look for an alternative: Free Pascal. The Free Pascal compiler is not yet 100is mature enough for commercial-grade software. thus came the idea of CrossFPC: A plugin for Delphi to compile Delphi code to Linux, and, as an added benefit: also for all other platforms, supported by Free Pascal.

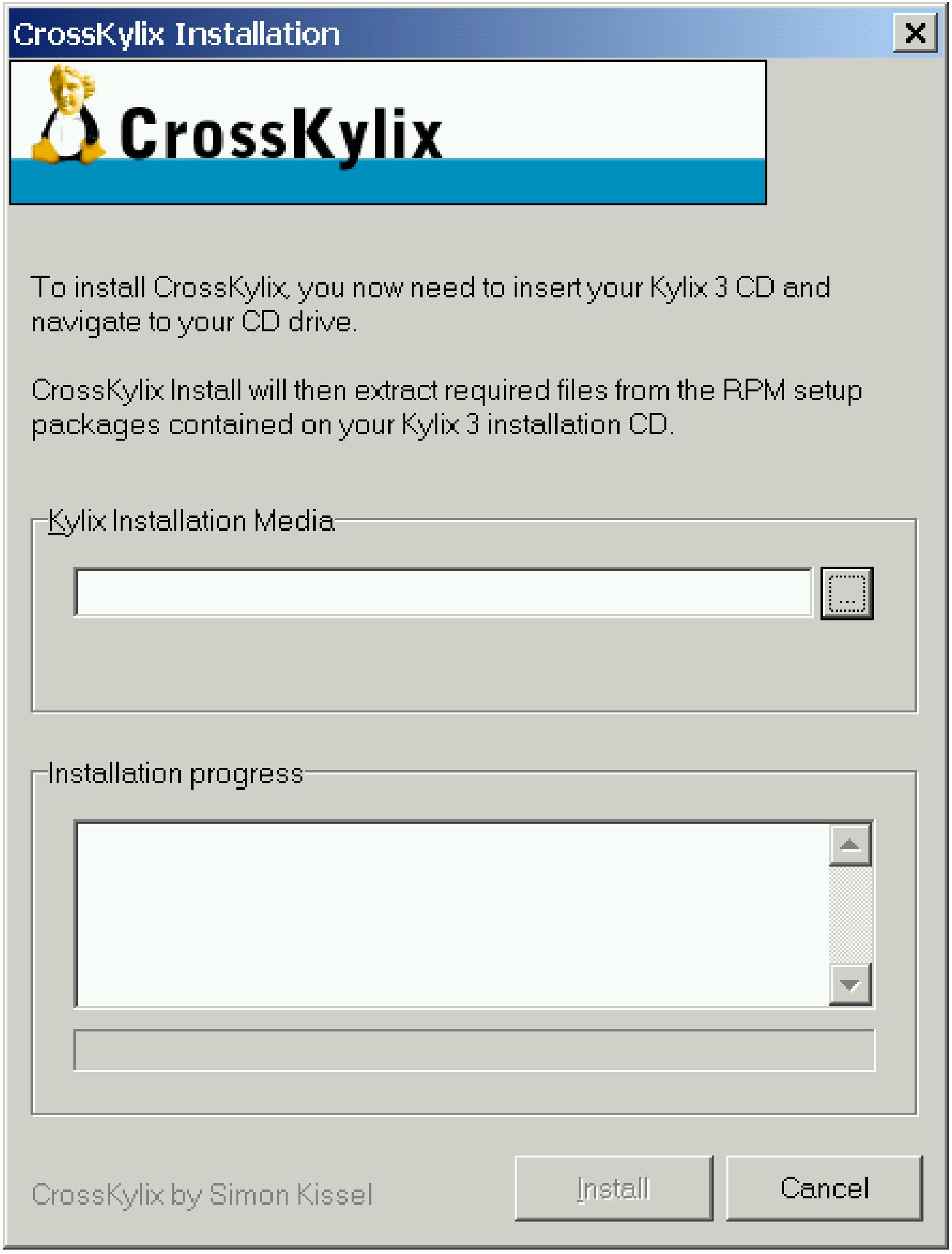
## 2 CrossKylix

Crosskylix is a small masterpiece: It installs itself as a small plugin in Delphi, where it sits as 3 menu items: 'Build with CrossKylix' and 'Compile with CrossKylix' under the 'Project' menu item, and 'CrossKylix' under the 'Tools' menu, for configuration. The latter allows to configure some extra directories which are available. But the nicest thing about CrossKylix is how it uses the Kylix compiler: The installation system takes the .rpm installation files from the Kylix installation CD-ROM and installs them wherever CrossKylix is installed (see figure 1 on page 2).

It also installs a minimal Linux emulator, which is able to run the Kylix compiler under Windows: It re-routes the Linux system calls to their Windows equivalents. Since the kylix binary is an Intel X86 binary, this is all that needs to be done.

That is all there is to it: Any CLX application (binary, package or library) can now be compiled from within Delphi by choosing the 'Project - Compile with CrossKylix' menu

Figure 1: CrossKylix install



CrossKylix Installation

To install CrossKylix, you now need to insert your Kylix 3 CD and navigate to your CD drive.

CrossKylix Install will then extract required files from the RPM setup packages contained on your Kylix 3 installation CD.

item. A dialog similar to the Delphi compiler progress dialog will appear. Messages will be shown in the message window, as expected.

It can be as simple as that. A simple plug-in, which takes away the hassle of re-booting your machine, copying files and correcting filenames (kylix is case-sensitive). And, Delphi itself still has numerous advantages over Kylix: Delphi 7 is much more stable than Kylix itself.

Besides this, CrossKylix also offers some unofficial patches to the LibC unit, and some patches which allow Kylix generated programs to run on newer distributions. This makes compiling and distributing Kylix programs a lot easier.

Unfortunately, Borland seems to have lost interest in Kylix. At least it has become very quiet around the FreeCLX and the Kylix compiler. Enter Free Pascal.

# 3 CrossFPC

Being concerned about the future of Kylix, the author of CrossKylix contacted the Free Pascal team, to cooperate on a new project: CrossFPC. In essence, the idea is quite simple: instead of using Kylix as the cross-compiler, use Free Pascal. Contrary to Kylix (or Delphi), Free Pascal is designed to be used as a cross-compiler. It can be used to compile Linux programs on Windows and vice versa. Or FreeBSD programs, or Mac OS X programs: it is not even restricted to the Intel processor.

On one side, this makes the task for the CrossFPC plugin easier: no linux emulation layer is needed. On the other hand, more files need to be distributed. Free Pascal relies on the GNU assembler (as) and linker (ld) to create the actual binaries: The compiler itself only generates assembler instructions (although an internal assembler exists). The initial (draft) version of CrossFPC comes with no less than 11 cross-compiling versions of the assembler and linker.

The setup of CrossFPC is thus easier than the setup for CrossFPC. Far mor difficult are some FPC-related issues: The problems to be solved are located in the compiler itself and the RTL:

- The FreeCLX sources assume certain routines to be present in the RTL.
- FPC does not yet support including compiled resources in a binary.
- Some additional variant support is needed.

These problems are worked on: The current CVS version of Free Pascal can compile the FreeCLX (except database routines).

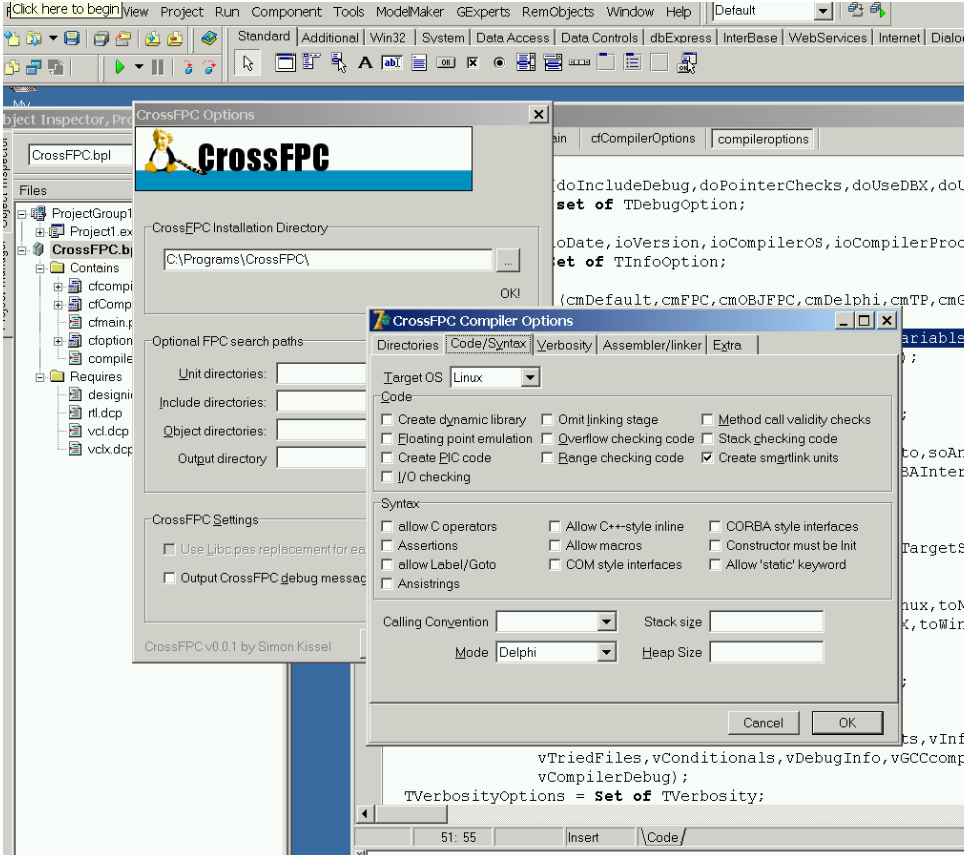
Inclusion of resources is being studied by the compiler team; A rudimentary schedule for resource inclusion is on the drawing board.

All the hooks for variants are in place; Some RTL routines still need to be filled in.

Until the resource linking issues are fixed, only command-line (console) programs can be cross-compiled from within Delphi, because the form definitions reside in a resource file. A possibility would be to use a Lazarus-like approach to tackle the resource loading issues: have a special resource management library handle all resources as string resources. However, this would require code changes to all files, and the aim is to have no special code changes needed for cross-fpc development.

The addition of a Free Pascal compiler options dialog (figure 2 on page 4) also allows Delphi to be used as an editor for FPC programs, and to compile the code from within Delphi - for Win32 or for any other platform.

Figure 2: CrossFPC options



CrossFPC Options

CrossFPC.bpl

CrossFPC

CrossFPC Installation Directory

C:\Programs\CrossFPC\ ...

OK!

Optional FPC search paths

Unit directories:

Include directories:

Object directories:

Output directory

CrossFPC Settings

□ Use Libc.pas replacement for ea

□ Output CrossFPC debug messag

CrossFPC v0.0.1 by Simon Kissel

CrossFPC Compiler Options

Directories | Code/Syntax | Verbosity | Assembler/linker | Extra

Target OS: Linux

Code

□ Create dynamic library
□ Floating point emulation
□ Create PIC code
□ I/O checking
□ Omit linking stage
□ Overflow checking code
□ Range checking code
□ Method call validity checks
□ Stack checking code
☑ Create smartlink units

Syntax

□ allow C operators
□ Assertions
□ allow Label/Goto
□ Ansistrings
□ Allow C++-style inline
□ Allow macros
□ COM style interfaces
□ CORBA style interfaces
□ Constructor must be Init
□ Allow 'static' keyword

Calling Convention:

Stack size:

Mode: Delphi

Heap Size:

Cancel | OK

```
[doIncludeDebug,doPointerChecks,doUseDBX,doU
 set of TDebugOption;

.oDate,ioVersion,ioCompilerOS,ioCompilerProc
Set of TInfoOption;

 (cmDefault,cmFPC,cmOBJFPC,cmDelphi,cmTP,cmG
ariabls
);

;

to,soAn
BAInter

TargetS

nux,toN
X,toWin

;

ts,vInf
              vTriedFiles,vConditionals,vDebugInfo,vGCCcomp
              vCompilerDebug);
  TVerbosityOptions = Set of TVerbosity;
```

## 4 Conclusion

At the moment of writing, the CrossFPC plugin is not yet ready to be deployed, but is functional. A first release is expected to be ready in the not-too-far future; When this happens, a link will be posted on the free pascal website. Hopefully, this will be realized when this article is published.

## 5 Web Resources

1. Crosskylix: http://crosskylix.untergrund.net/
2. Free Pascal: http://www.freepascal.org/