

ReactOS Build Environment

Version 2.1.2 for Unix-based operating systems

Base package and i386 build tools

by Pierre Schweitzer (pierre@reactos.org)



Introduction

The ReactOS Build Environment (“RosBE”) is a Build Environment for building ReactOS. This version is targeted at Unix-based operating systems such as Linux or FreeBSD.

The Installation process of the Build Environment is divided into two parts.

First a script builds the needed compiler tools for your operating system and installs them. Therefore the “gcc” compiler for your system and some other build tools have to be installed.

Then another script creates a shortcut for you. This shortcut opens a Terminal window and prepares all environment settings for building ReactOS.

Needed Applications

Make sure that the following applications are installed on your system. They are needed to build the compiler tools. The “RosBE-Builder.sh” script automatically checks for them:

- **GNU Bash**
The executable file and the package name are often called “bash”. All scripts are designed to work only under the Bash shell. Please also make sure that the file “/bin/bash” exists. For example FreeBSD usually installs Bash into another directory, so you have to create a symbolic link “/bin/bash” to the executable file.
- **GNU Bison**
The executable file and the package name are often called “bison”.
- **Flex**
The executable file and the package name are often called “flex”.
- **GNU GCC**
You need a GCC compiler, which is suitable for your system. The executable file and the package name are often called “gcc”. Your GCC compiler needs to be able to compile both C and C++ sources. Therefore the “g++” module also has to be installed. The compiler needs to be in version 3.4 or later.
- **Grep**
The executable file and the package name are often called “grep”.
- **GNU Make**
Make sure that you have the GNU version of “make” installed. For example FreeBSD uses another version. The executable file and the

package name are often called “make” or “gmake”.

- GNU Makeinfo
The executable file is often called “makeinfo”. Usually this file is in a package called “texinfo”.

Needed Libraries

Make sure that the following libraries are installed on your system. They are needed to extract the compiler tools. The “RosBE-Builder.sh” script automatically checks for them:

- zlib
The library file is generally called “libz.so”.

Preparing the installation

Under some operating systems, you need to perform additional steps before you can install the Build Environment. This section lists them.

- FreeBSD
 - You have to create a symbolic link “/bin/bash”, because FreeBSD installs Bash to “/usr/local/bin/bash”. Simply execute the following command for doing this:

```
In -s /usr/local/bin/bash /bin/bash
```

Installation

The Installation of RosBE is very simple, since there are scripts, which do most of the work for you. Follow these steps to compile and install RosBE for your used operating system:

1. Open a Terminal window and type “su” to become “root” user.
NOTE: Becoming a root user is only required for installing RosBE into the default directory (“/usr/local/RosBE”). If you want to install it into a personal directory, you don't need superuser privileges.
2. Call the “RosBE-Builder.sh” script in the directory of the unpacked RosBE files. **NOTE:** Advanced users can make use of CFLAGS, CC, CXX variables to modulate the way tools will be built.
3. Follow the directions on the screen until the script ends with “Finished successfully!”
4. Type “exit” to switch back to your normal user account. Then call the “createshortcut.sh” script in the installation directory (usually this is “/usr/local/RosBE”).
5. Follow the directions on the screen to create a shortcut for the Build Environment.

After that you can easily open a Build Environment for ReactOS by opening the shortcut.

Uninstallation

If you want to uninstall RosBE, just delete the entire installation directory and any shortcuts you have created.

Version Information

This version of RosBE for Unix-based operating systems uses the following components:

- GNU Binutils v2.23
compiled for mingw32
patched with:
 - <http://svn.reactos.org/project-tools/trunk/RosBE/Patches/binutils-changes-against-2.22.patch?p=1352>
 - <http://sourceware.org/ml/binutils/2012-10/msg00061.html>
 - <http://sourceware.org/ml/binutils/2012-10/msg00062.html>
- GCC, The GNU Compiler Collection v4.7.1
compiled for mingw32
patched with:
 - http://gcc.gnu.org/viewcvs/gcc/branches/gcc-4_7-branch/gcc/ira-int.h?view=patch&r1=191605&r2=191604&pathrev=191605
 - <https://github.com/DragonFlyBSD/DPorts/commit/a680cc6ef758e2f15be8bf8209da51658e02d710>
 - <https://gcc.gnu.org/ml/gcc-patches/2014-05/msg01401.html>

NOTE: This GCC version only supports C and C++
- Make 3.82
patched with:
 - <http://svn.reactos.org/project-tools/trunk/RosBE/Patches/Make-Windows-jobserver-against-3.8.2.patch?p=1321>
- CMake 3.2.1-ReactOS
- GMP 5.0.5
- Ninja 1.6.0
- MinGW-Runtime 3.17
- MPFR 3.1.1
- W32api 3.14

Tested Operating Systems

The Build Environment was designed to work on all Unix-based operating systems. Nevertheless, it is impossible to test it on all possible systems out there.

The current primary development and test system is Ubuntu 14.04. But users

also reported that it works well under other Linux systems such as Fedora or openSUSE and Unix derivatives like FreeBSD or Mac OS X.

You should be able to compile the build tools on both i386 (32-bit) and x86_64 (64-bit) systems.

Websites

MinGW – Minimalist GNU for Windows

<http://www.mingw.org>

Binutils, the Binary Tools Collection

<http://www.gnu.org/software/binutils>

CMake, Cross Platform Make

<http://www.cmake.org>

GCC, the GNU Compiler Collection

<http://www.gnu.org>

GMP, the GNU MP Bignum library

<http://www.gmp.org>

Make, the Makefile processor

<http://www.gnu.org/software/make>

The GNU MPFR library

<http://www.mpfr.org>

Credits

Colin Finck (colin@reactos.org)

Pierre Schweitzer (pierre@reactos.org)

Thanks go also to the members of the RosBE-Windows team, namely:

Daniel Reimer (daniel.reimer@reactos.org)

Peter Ward (dralnx@gmail.com)

Special thanks to ROSFan, who created the nice ReactOS Build Environment Logo.

Special thanks also goes out to all those who provide so much help by testing out new versions, you know who you are. Without them the RosBE would not be what it is.

Credits go to Gregor Anich who created the original ReactOS Build Environment v0.2-3.4.4.

Inspired by Dazzle from TinyKRNL.