EBImage installation HOWTO

Oleg Sklyar

osklyar@ebi.ac.uk

October 22, 2008

The package was tested to work on Linux (32/64 bit), MacOS, Windows XP (32 bit) and Solaris (64 bit). It should be possible to install and run the package on other versions of UNIX-like and Windows systems in a similar manner. On Windows, all dependencies are required. On other systems, some dependencies are recommended (but not required) in order to enjoy the full functionality of the package. GTK+ support was not tested on MacOS, but there is no apparent reason for the package to fail to support it.

1 UNIX/Linux Installation

The installation should be performed from the source after the following dependencies are satisfied:

ImageMagick version at least 6.2.0,

GTK+ version at least 2.6 (recommended).

On most Linux distributions, these libraries are available as precompiled binaries in `rpm` or `deb` format. You will also need the corresponding header files[1], these are in equinomous package with suffix `-dev` or `-devel`. The following instructions work for Ubuntu Linux, but other distributions have similar packages with slight naming differences.

For ImageMagick, you will need to install `libmagick9-dev` (including its dependencies) using `apt-get` or the Synaptic package manager. Alternatively, you can download the latest version of ImageMagick from `http://www.imagemagick.org` and install it from source using

```
./configure && make && make install
```

---

[1] A library's header files are necessary to compile other programs, such as EBImage, that use the library. The header files contain necessary information about the functionality of the library.

This will also install the header files.

EBImage will function without GTK+, but most of its GUI functionality (image chooser, interactive image display with zoom features etc) will be disabled. You will need `libgtk2.0-dev` with all the dependencies, which normally include `glib`, `pango-dev`, `libcairo-dev`, `atk-dev` and some others. Alternatively, all these packages can be installed from source available at `http://www.gtk.org` along with the corresponding installation instructions. When installing GTK+, please ensure that you also install `pkg-config` as it is used to determine the location of GTK+ include files and its compile flags. Without this package, GTK+ will not be located on your system by the package installer. `pkg-config` is normally installed in UNIX/Linux systems running the GNOME desktop manager, it is also likely in this case that you have necessary GTK+ development files because GNOME is build upon GTK+.

After the dependencies are installed, install the package with the standard `R CMD INSTALL` script.

2 MacOS Installation

Analogously to the UNIX/Linux installation, this should be performed from source using the same technique and dependencies.

One can use the fink project (`http://fink.sf.net`) to download and install dependencies, or install them from source as in Linux. Both work fine. Sometimes, MacOS users get problems with a UNIX `libtool`, it is likely that a MacOS-specific `libtool` should be installed, which is named `glibtool`. It is also available from fink.

I have not yet checked if the GTK+-enabled version works on MacOS, but since `gtk+2-dev` and `gtk+2-shlib` are available at fink, it is worth trying. Please note that you will need `pkgconfig` (also available from fink) in order for the installation script to find your GTK+ includes.

3 Windows 32bit Installation

The Windows binary package (extension `.zip`) contains a DLL that was compiled on a 32 bit Windows XP machine. The source package (extension `.tar.gz`) is distributed with a *precompiled* Windows DLL included in `.\inst\winlibs\`. This file is moved to the package's `.\libs\` during the Windows build process (`R CMD build -binary`) avoiding the compilation. If you do want to recompile the source package on Windows, please delete the `.\inst\winlibs\` directory from the source tarball.

Whether you are installing a binary package or compiling it from source on Windows, you will need to have *both* ImageMagick and GTK+ installed and *added* to your or system-wide `PATH` variable prior to the package installation.

If you plan to compile the package from source, first follow the R instructions on compiling Windows packages and install all the necessary tools (Cygwin, MSYS, MinGW, MS Help Compiler etc). In the latter case it is recommended to install MinGW and MSYS into `c:\mingw` as this path is hard coded into `.\src\Makevars.win` of the package. Otherwise you will need to edit this file. You do not need MinGW or MSYS if you install the binary package. For the binary package, just install the dependencies so that the library could function.

I have not checked if the package compiles on 64 bit Windows since I don't have such hardware.

3.1 Installing ImageMagick

The Windows version of ImageMagick is also available from `http://www.imagemagick.org`. Please ensure that you download a package named like `ImageMagick-6.x.x-x-Q16-windows-dll.exe`, where `x.x-x` stand for the minor version number. You need to pay attention that you download `*-windows-dll.exe` and not `*-static.exe` otherwise the shared libraries to use with the package will not be installed. You also should use the version with 16 bit `QUANTUM` depth (`*-Q16-*`) to ensure the best possible color range.

    After starting the installation wizard, you can select the default settings and let the installer to register environmental variables for you. Ensure that the path to ImageMagick was added to your `PATH` variable, e.g. that it contains somthing like `c:\ImageMagick;`.

    If you are planning to compile EBImage from source, select during the installation that you want development files to be installed as well of. In this case, it is strongly recommended to install ImageMagick into a path that does not contain spaces. The recommended, and hard coded into `.\src\Makevars.win` is `c:\ImageMagick`. If select another location, please be sure to update this file.

3.2 Installing GTK+

It is recommended to use the GladeWin32 project, `http://gladewin32.sf.net` for the source of GTK+ installer. The `www.gtk.org` site has its own installers, but the above one is an all-in-one installation which perfectly fits the needs of EBImage. Please download the latest available release of *Gtk+/Win32 Development Environment* if you are planning to recompile the package from source, or *Gtk+/Win32 Runtime Environment* if you are just planning to use the binary. For the binary installation just run the installer and accept the default suggestions: the installer will register all environmental variables for you.

    Please ensure now that the installer added the path to GTK+ `\bin` directory *before* the path to `c:\Windows\` and `c:\Windows\System32`. Otherwise, the package might find the wrong `xmlparser.dll` which is included in both, GTK+ and Windows itself (and is significantly outdated in Windows).

    If you are planning to compile EBImage from source, install GTK+ into a path that does not contain spaces. It is strongly recommended to use the same directory where you installed MSYS and MinGW, namely `c:\mingw`. This directory is hard coded into `.\src\Makevars.win` and you will need to modify this file if installing GTK+ elsewhere.

3.3 Installing the package

The binary package can be installed now using the R menu command to install a package from a local file.

    To compile from source, ensure that R is in your `PATH`, run `cmd` and switch to the directory with the package tarball. Now run `R CMD build -binary` to generate the binary. Then install it as above.