

Using *zlibbioc*

Martin Morgan

Modified: 29 September, 2011. Compiled: October 27, 2020

The *zlibbioc* package is meant as a utility for package developers. It contains the source code to the zlib library, and can be used to access zlib shared library functionality. The library is made available as `libzbioc`.

The *zlibbioc* package is installed in the normal R manner. The `libzbioc` library is always built on Windows, but on other platforms it is only built when provided with the configure option `--with-libzbioc`, e.g., as

```
R CMD INSTALL --configure-args="--with-libzbioc" zlibbioc_<...>.tar.gz
```

or

```
> install.pacakges("zlibbioc_<...>.tar.gz",  
                   configure.args="--with-libzbioc")
```

MacOS has `zlib` installed, so building the libraries are neither necessary nor supported on that platform. Advanced use cases may require consultation of instructions in `zlibbioc/src/zlib-1.2.5/configure`.

All packages wishing to use the libraries in *zlibbioc* must

- Add Imports: `zlibbioc` to the DESCRIPTION file.
- Add `import(zlibbioc)` to the NAMESPACE file.

Reference the relevant include file in your C source code:

```
#include "zlib.h"
```

The content of the include files can be found in the *zlibbioc* source (under `src/zlib-1.2.5`) or at their installed location.

On Windows, the recommended approach is to link to the DLL. This requires that the appropriate header files are available to the gcc compiler, and that the DLL is discovered by the linker.

- Create a file `src/Makevars.win` including the following lines:

```
ZLIB_CFLAGS+=$(shell echo 'zlibbioc::pkgconfig("PKG_CFLAGS")' |\n  "${R_HOME}/bin/R" --vanilla --slave)\nPKG_LIBS+=$(shell echo 'zlibbioc::pkgconfig("PKG_LIBS_shared")' |\n  "${R_HOME}/bin/R" --vanilla --slave)
```

```
%.o: %.c
    $(CC) $(ZLIB_CFLAGS) $(ALL_CPPFLAGS) $(ALL_CFLAGS) -c $< -
o $@
```

Packages with C++ code also require the rule (replace `.cc` with `.cpp` as necessary)

```
%.o: %.cc
    $(CXX) $(ZLIB_CFLAGS) $(ALL_CPPFLAGS) $(ALL_CXXFLAGS) -c $< -
o $@
```

(remember that the second line of each rule begins with a tab, not spaces).

On Linux and other platforms, the most portable solution is to link to static libraries

- Create a file `src/Makevars` including the following lines:

```
PKG_CFLAGS+=$(shell echo 'zlibbioc::pkgconfig("PKG_CFLAGS")' |\
    "${R_HOME}/bin/R" --vanilla --slave)
PKG_LIBS+=$(shell echo 'zlibbioc::pkgconfig("PKG_LIBS_static")' |\
    "${R_HOME}/bin/R" --vanilla --slave)
```

It is also possible to link to the shared library (see qualifications about portability in ‘Writing R Extensions’) with

```
PKG_CFLAGS+=$(shell echo 'zlibbioc::pkgconfig("PKG_CFLAGS")' |\
    "${R_HOME}/bin/R" --vanilla --slave)
PKG_LIBS+=$(shell echo 'zlibbioc::pkgconfig("PKG_LIBS_shared")' |\
    "${R_HOME}/bin/R" --vanilla --slave)
```

The *Rsamtools* package is a more complex example illustrating this approach.