Package: rvtk (via r-universe)

May 24, 2026

Title Bindings for the Visualization Toolkit ('VTK')

Version 0.2.0

Description Provides pre-compiled static 'VTK' libraries and headers so that downstream R packages can link against the Visualization Toolkit without requiring users to install 'VTK' manually. On all platforms the package first honours a user-supplied 'VTK_DIR' environment variable. On macOS it then tries 'Homebrew', followed by 'pkg-config'. On Linux it tries 'pkg-config' and well-known system prefixes ('/usr', '/usr/local'). If no suitable system installation is found on macOS or Linux, pre-built static libraries are downloaded automatically from the package's GitHub releases for x86_64 and aarch64. On Windows the package tries 'VTK_DIR', then 'Rtools45' 'pacman', then common 'MSYS2' prefixes, accepting both static ('.a') and shared ('.dll.a' import libs + DLLs) installations. When shared libraries are used, the VTK DLLs are staged in 'inst/vtk-dlls/' and an '.onLoad' hook prepends that directory to PATH via 'Sys.setenv()' when the package is loaded, and restored in '.onUnload()'. The pre-built fallback downloads static libraries by default; set 'VTK_LINK_TYPE=shared' before installation to download the DLL build instead. The pre-built bundles include the following VTK modules (plus their transitive dependencies): 'VTK_IOLegacy', 'VTK_IOXML', 'VTK_IOCore', 'VTK_CommonCore', and 'VTK_CommonDataModel'. Note that on Windows 'VTK_IONetCDF', 'VTK_IOHDF', 'VTK_GeovisCore', and 'VTK_RenderingCore' are additionally disabled because 'netcdf' and 'libproj' are not available in the 'Rtools45' 'static.posix' sysroot. Downstream packages can declare 'Imports: rvtk' and obtain the correct compiler and linker flags at install time via rvtk::CppFlags() and rvtk::LdFlagsFile(), optionally restricting linking to a subset of modules via the 'modules' argument. A convenience helper use_rvtk() (in the spirit of the 'usethis' package) automates the full setup of a downstream package: it adds 'rvtk' to 'Imports' in 'DESCRIPTION' and writes all files required to resolve VTK flags at install time ('src/Makevars',

'src/Makevars.win', 'tools/configure.R', and 'R/rvtk_imports.R'), and keeps generated build artefacts out of version control.

License MIT + file LICENSE

Encoding UTF-8

URL https://github.com/astamm/rvtk, https://astamm.github.io/rvtk/

BugReports https://github.com/astamm/rvtk/issues

NeedsCompilation no

Suggests cli, tinytest

Config/roxygen2/version 8.0.0

Config/roxygen2/markdown TRUE

Repository https://astamm.r-universe.dev

Date/Publication 2026-05-24 14:28:35 UTC

RemoteUrl https://github.com/astamm/rvtk

RemoteRef HEAD

RemoteSha ff412edd17fd7dd9ed5c0a9e0c9b1f66f5a9923c

Contents

`CppFlags` *Compiler flags for packages linking against VTK*

Description

Returns the C pre-processor flags (`-I` paths) required to compile C++ code that includes VTK headers. Intended to be called from a downstream package's `configure` or `configure.win` script:

Usage

```
CppFlags()
```

Details

```
VTK_CPPFLAGS="$("${R_HOME}/bin/Rscript" --vanilla -e "rvtk::CppFlags()")"
```

Value

A single character string of compiler flags, written to stdout (so that it can be captured by shell command substitution in `configure`) and returned invisibly.

Examples

```
flags <- CppFlags()
```

`LdFlags` *Linker flags for packages linking against VTK*

Description

Returns the linker flags (`-L` paths and `-l` library names) required to link C++ code against VTK. Intended to be called from a downstream package's `configure` or `configure.win` script:

Usage

```
LdFlags(modules = NULL)
```

Arguments

`modules`	A character vector of VTK module names to link against, e.g. `c("vtkIOLegacy", "vtkCommonCore")`. When `NULL` (the default) all available modules are included. When non-`NULL`, only the requested modules are linked, which avoids pulling in unneeded symbols (such as AppKit/Cocoa symbols from rendering modules when using the pre-built static bundle).

Details

```
VTK_LIBS="$("${R_HOME}/bin/Rscript" --vanilla -e "rvtk::LdFlags()")"
```

On Windows the full set of VTK linker flags can exceed the 8 191-character command-line limit. Prefer `LdFlagsFile()` on Windows to write the flags to a response file instead.

Value

A single character string of linker flags, written to stdout (so that it can be captured by shell command substitution in `configure`) and returned invisibly.

Examples

```
flags <- LdFlags()
```

`LdFlagsFile` *Write VTK linker flags to a response file*

Description

On Windows the full set of VTK linker flags can exceed the 8 191-character Windows command-line limit, causing the linker to drop flags at the end of the list. This function writes the flags to a plain-text response file that the linker reads via the `@file` syntax, keeping the command line short.

Usage

```
LdFlagsFile(path, modules = NULL, os_type = .Platform$OS.type)
```

Arguments

`path`	Path (relative to the package source root, i.e. where `configure` runs) to the response file to write on Windows, e.g. `"src/vtk_libs.rsp"`. Ignored on non-Windows platforms.
`modules`	A character vector of VTK module names to link against, e.g. `c("vtkIOLegacy", "vtkCommonCore")`. When `NULL` (the default) all available modules are included. When non-`NULL`, only the requested modules are linked, which avoids pulling in unneeded symbols (such as AppKit/Cocoa symbols from rendering modules when using the pre-built static bundle).
`os_type`	A string identifying the operating-system type, defaulting to `.Platform$OS.type`. Override to `"windows"` or `"unix"` in tests to exercise the Windows response-file branch without needing a Windows environment.

Details

Intended to be called from a downstream package's `configure` or `configure.win` script:

```
VTK_LIBS="$("${R_HOME}/bin/Rscript" --vanilla -e \
  "rvtk::LdFlagsFile('src/vtk_libs.rsp')")"
# VTK_LIBS is now the short string "@src/vtk_libs.rsp" on Windows,
# or the raw flags on macOS/Linux.
```

On Windows the flags are written to `path` and the function returns the `@basename(path)` token for the linker. On macOS and Linux, `ld` does not reliably support `@file` response files at the compiler-driver level, so no file is written and the raw flags are returned directly.

Value

Invisibly, the string to embed in `configure` (either `@basename(path)` on Windows or the raw flags on other platforms). The string is also written to stdout so that shell command substitution captures it.

Examples

```
rsp <- file.path(tempdir(), "vtk_libs.rsp")
ref <- LdFlagsFile(rsp)
```

`use_rvtk` *Set up a downstream package to use rvtk*

Description

A usethis-style helper that configures a downstream R package to link against VTK via **rvtk**. It performs the following steps:

- Adds `rvtk` to the `Imports` field of `DESCRIPTION`.
- Writes `src/Makevars` that queries compiler and linker flags at install time by calling `tools/configure.R`.
- Writes `src/Makevars.win` with the Windows-specific `$(shell ...)` syntax that does the same.
- Writes `tools/configure.R` that calls `CppFlags()` and `LdFlagsFile()` with the requested VTK modules.
- Adds `src/vtk_libs.rsp` to `.gitignore` (it is generated at install time and must not be committed).
- Creates `R/rvtk_imports.R` with a minimal `@importFrom rvtk` roxygen tag so that `R CMD check` does not complain about **rvtk** being listed in `Imports` without any function import in the R code.

After running `use_rvtk()` the downstream package is fully configured: VTK compiler and linker flags are resolved automatically at `R CMD INSTALL` time on all platforms without any shell `configure` / `configure.win` scripts.

Usage

```
use_rvtk(
 modules = c("vtkIOLegacy", "vtkIOXML", "vtkIOXMLParser", "vtkIOCore", "vtkCommonCore",
  "vtkCommonDataModel", "vtkCommonExecutionModel", "vtkCommonMath", "vtkCommonMisc",
   "vtkCommonSystem", "vtkCommonTransforms", "vtksys"),
  path = "."
)
```

Arguments

`modules`	A character vector of VTK module names to link against. These are passed to `LdFlagsFile()` in the generated `tools/configure.R`, restricting linking to only the modules the downstream package needs. Defaults to a standard set covering common I/O and core modules (the same set used by the reference implementation in https://github.com/tractoverse/riot).
`path`	Path to the root of the downstream package. Defaults to the current working directory.

Value

Invisibly, the normalised path to the package root. Called primarily for its side effects.

See Also

`CppFlags()`, `LdFlagsFile()`

`VtkVersion` *VTK version used by this package*

Description

VTK version used by this package

Usage

```
VtkVersion()
```

Value

A character string with the VTK version, e.g. `"9.3.1"`.

Examples

```
VtkVersion()
```

Index