Package: rvtk (via r-universe)

May 11, 2026

Title Bindings for the Visualization Toolkit ('VTK')

Version 0.1.3

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. 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. Note that on Windows the modules 'VTK_IONetCDF', 'VTK_IOHDF', 'VTK_GeovisCore', and 'VTK_RenderingCore' are 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().

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.3

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

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

NeedsCompilation no

Suggests tinytest

Author Aymeric Stamm [aut, cre] (ORCID:
<https://orcid.org/0000-0002-8725-3654>)

Maintainer Aymeric Stamm <aymeric.stamm@cnrs.fr>

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

Date/Publication 2026-05-11 08:39:59 UTC

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

RemoteRef HEAD

RemoteSha 688f527eaa01a4a8886732734af6240ad460b0c1

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()
```

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, 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.
`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)
```

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