

Package: gdalraster.windows (via r-universe)

October 18, 2026

Title Self-Contained gdalraster Builds for Windows

Version 0.5.0

Description Installs a self-contained GDAL build runtime on Windows and builds a vendored gdalraster package against it. The resulting package carries its native dependency closure and runtime data, so it loads in fresh R sessions without activation hooks or environment changes.

License MIT + file LICENSE

URL <https://github.com/jimbrig/gdalraster.windows>,
<https://docs.jimbrig.com/gdalraster.windows/>

BugReports <https://github.com/jimbrig/gdalraster.windows/issues>

Imports cli, gitcreds, http2, pkgbuild, rlang, stats, tools, utils,
withr

Suggests gdalraster, knitr, quarto, testthat (>= 3.0.0)

VignetteBuilder quarto

Config/testthat/edition 3

Encoding UTF-8

Language en-US

Roxygen list(markdown = TRUE)

Config/roxygen2/version 8.1.0

Config/pak/sysreqs git libssl-dev

Repository <https://jimbrig.r-universe.dev>

Date/Publication 2026-08-18 23:59:40 UTC

RemoteUrl <https://github.com/jimbrig/gdalraster.windows>

RemoteRef HEAD

RemoteSha f3f8c014a008c96f686b7948f376b2fbbaa20296e

Contents

gdal_build_gdalraster	2
gdal_enable_python	3
gdal_home	4
gdal_install_runtime	4
gdal_setup	5
gdal_sitrep	6
gdal_uninstall	7
gdal_update	7
gdal_verify	8
Index	10

gdal_build_gdalraster	<i>Build a self-contained gdalraster package</i>
-----------------------	--

Description

Compiles gdalraster against the managed GDAL SDK, vendors the bundle’s DLL dependency closure, GDAL/PROJ data, and pure-Python utilities into the installed package, then records build provenance.

Usage

```
gdal_build_gdalraster(  
    gdal_home = default_gdal_home(),  
    lib = NULL,  
    isolated = FALSE,  
    user_lib = NULL,  
    source_tarball = NULL,  
    repo = .gdalraster_repo,  
    ref = "HEAD",  
    upgrade = FALSE,  
    repos = getOption("repos"),  
    force = FALSE,  
    enable_python = TRUE  
)
```

Arguments

gdal_home	Installed GDAL build runtime.
lib	Destination library. Defaults to .libPaths()[1].
isolated	Install into the package-managed isolated library instead of .libPaths()[1]. Ignored when lib is set.
user_lib	Deprecated. user_lib = TRUE is now the default; user_lib = FALSE is equivalent to isolated = TRUE.

source_tarball	Optional local gdalraster source tarball.
repo	Upstream source repository.
ref	Git reference used for the source archive.
upgrade	Install missing R dependencies before compiling.
repos	CRAN-like repositories used when upgrade = TRUE.
force	Replace an existing gdalraster package.
enable_python	Provision the managed embedded-Python .pth file after installation when possible.

Value

Invisibly, the destination library.

gdal_enable_python	<i>Enable GDAL embedded-Python utilities</i>
--------------------	--

Description

Writes a managed .pth file into the system CPython site-packages directory. The file points to the pure-Python osgeo_utils vendored inside the installed gdalraster package. This does not modify PYTHONPATH and does not affect isolated virtual environments.

Usage

```
gdal_enable_python(
  lib = NULL,
  isolated = FALSE,
  user_lib = NULL,
  python = NULL,
  site_packages = NULL,
  quiet = FALSE
)
```

Arguments

lib	Library containing the self-contained gdalraster package. Defaults to .libPaths()[1].
isolated	Use the package-managed isolated library instead of .libPaths()[1]. Ignored when lib is set.
user_lib	Deprecated. user_lib = TRUE is now the default; user_lib = FALSE is equivalent to isolated = TRUE.
python	Optional path to the CPython executable GDAL embeds.
site_packages	Optional explicit site-packages directory.
quiet	Suppress status messages.

Value

Invisibly, the .pth path, or NULL when no system CPython can be located.

gdal_home	<i>Resolve the managed GDAL runtime directory</i>
-----------	---

Description

Resolution order is the `gdalraster.windows.gdal_home` option, the `GDALRASTER_WINDOWS_GDAL_HOME` environment variable, then the package-managed user data directory. The returned path is normalized to forward slashes.

Usage

`gdal_home()`

Value

A single path.

gdal_install_runtime	<i>Install a self-contained GDAL build runtime</i>
----------------------	--

Description

Installs a release bundle used to compile `gdal_build_gdalraster()`. The installed runtime is a build-time SDK; packages produced by `gdal_build_gdalraster()` vendor everything needed at run time.

Usage

```
gdal_install_runtime(  
    repo = .bundle_repo,  
    tag = "latest",  
    asset_pattern = .bundle_asset_pattern,  
    gdal_home = default_gdal_home(),  
    force = FALSE,  
    local_zip = NULL,  
    fallback_zip = NULL  
)
```

Arguments

- | | |
|----------------------------|---|
| <code>repo</code> | GitHub repository that publishes bundle releases. |
| <code>tag</code> | Bundle release tag or "latest". |
| <code>asset_pattern</code> | Regular expression selecting a bundle zip asset. |
| <code>gdal_home</code> | Destination runtime directory. |
| <code>force</code> | Reinstall an existing runtime. |
| <code>local_zip</code> | Optional local bundle zip. This takes precedence over a release download. |
| <code>fallback_zip</code> | Optional local zip used if release resolution or download fails. |

Details

The installer is idempotent. When the requested bundle tag is already recorded in MANIFEST.dcf, installation is skipped unless force = TRUE.

Value

Invisibly, the installed runtime directory.

gdal_setup	<i>Set up a self-contained gdalraster installation</i>
------------	--

Description

Installs the GDAL build runtime when needed, builds and vendors gdalraster, provisions embedded Python when available, and verifies the result in fresh processes. The default destination is .libPaths()[1].

Usage

```
gdal_setup(  
  lib = NULL,  
  isolated = FALSE,  
  user_lib = NULL,  
  update = FALSE,  
  force = FALSE,  
  tag = "latest",  
  local_zip = NULL,  
  fallback_zip = NULL,  
  source_tarball = NULL,  
  upgrade = FALSE,  
  verify = TRUE  
)
```

Arguments

lib	Destination library for gdalraster. Defaults to .libPaths()[1].
isolated	Install into the package-managed isolated library instead of .libPaths()[1]. Ignored when lib is set.
user_lib	Deprecated. user_lib = TRUE is now the default; user_lib = FALSE is equivalent to isolated = TRUE.
update	Upgrade the runtime to the requested/latest bundle and rebuild.
force	Reinstall and rebuild even when provenance is current.
tag	Runtime bundle release tag or "latest".
local_zip	Optional local runtime bundle.
fallback_zip	Optional local fallback runtime bundle.

source_tarball	Optional local gdalraster source tarball.
upgrade	Install gdalraster R dependencies before compiling.
verify	Run <code>gdal_verify()</code> after setup.

Value

Invisibly, a list describing performed actions.

gdal_sitrep	<i>Report GDAL and gdalraster installation state</i>
-------------	--

Description

Reports the installed GDAL runtime, the gdalraster build in the active library, staleness against the newest gdal-v* bundle, embedded-Python provisioning, and session collision risks without changing the session.

Usage

```
gdal_sitrep(
  lib = NULL,
  isolated = FALSE,
  user_lib = NULL,
  network = rlang::is_interactive(),
  quiet = FALSE
)
```

Arguments

lib	Library expected to contain the managed gdalraster build. Defaults to <code>.libPaths()[1]</code> .
isolated	Inspect the package-managed isolated library instead of <code>.libPaths()[1]</code> . Ignored when lib is set.
user_lib	Deprecated. <code>user_lib = TRUE</code> is now the default; <code>user_lib = FALSE</code> is equivalent to <code>isolated = TRUE</code> .
network	Query GitHub for the latest GDAL runtime bundle (gdal-v*).
quiet	Return state without printing it.

Details

The default inspects `.libPaths()[1]`, the library used by `library(gdalraster)`. Pass `lib` for a custom library, or `isolated = TRUE` for the package-managed isolated library.

Value

Invisibly, a named list describing installation state.

gdal_uninstall	<i>Uninstall managed GDAL resources</i>
----------------	---

Description

Uninstall managed GDAL resources

Usage

```
gdal_uninstall(  
    lib = NULL,  
    isolated = FALSE,  
    user_lib = NULL,  
    runtime = TRUE,  
    package = TRUE,  
    python = TRUE,  
    force = FALSE  
)
```

Arguments

lib	Library containing the managed gdalraster package. Defaults to <code>.libPaths()[1]</code> .
isolated	Remove the package-managed isolated library instead of <code>.libPaths()[1]</code> . Ignored when lib is set.
user_lib	Deprecated. <code>user_lib = TRUE</code> is now the default; <code>user_lib = FALSE</code> is equivalent to <code>isolated = TRUE</code> .
runtime	Remove the GDAL build runtime.
package	Remove the managed gdalraster package.
python	Remove managed embedded-Python <code>.pth</code> files.
force	Confirm removal in non-interactive sessions.

Value

Invisibly, the removed paths.

gdal_update	<i>Update the GDAL runtime and rebuild gdalraster</i>
-------------	---

Description

Update the GDAL runtime and rebuild gdalraster

Usage

```
gdal_update(
  lib = NULL,
  isolated = FALSE,
  user_lib = NULL,
  tag = "latest",
  local_zip = NULL,
  fallback_zip = NULL,
  source_tarball = NULL,
  upgrade = FALSE,
  verify = TRUE
)
```

Arguments

lib	Destination library for gdalraster. Defaults to <code>.libPaths()[1]</code> .
isolated	Install into the package-managed isolated library instead of <code>.libPaths()[1]</code> . Ignored when lib is set.
user_lib	Deprecated. <code>user_lib = TRUE</code> is now the default; <code>user_lib = FALSE</code> is equivalent to <code>isolated = TRUE</code> .
tag	Runtime bundle release tag or "latest".
local_zip	Optional local runtime bundle.
fallback_zip	Optional local fallback runtime bundle.
source_tarball	Optional local gdalraster source tarball.
upgrade	Install gdalraster R dependencies before compiling.
verify	Run <code>gdal_verify()</code> after setup.

Value

The result of `gdal_setup()`, invisibly.

gdal_verify

Verify the self-contained gdalraster installation

Description

Runs verification in fresh Rscript `--vanilla` processes. Checks the GDAL Algorithm API, GEOS and CRS support, Arrow/Parquet/HDF5/netCDF driver registration, a first Parquet dataset open, and the GeoPackage Python validator when embedded Python has been provisioned.

Usage

```
gdal_verify(  
    lib = NULL,  
    isolated = FALSE,  
    user_lib = NULL,  
    python = TRUE,  
    quiet = FALSE  
)
```

Arguments

lib	Library containing the managed gdalraster package. Defaults to <code>.libPaths()[1]</code> .
isolated	Verify the package-managed isolated library instead of <code>.libPaths()[1]</code> . Ignored when lib is set.
user_lib	Deprecated. <code>user_lib = TRUE</code> is now the default; <code>user_lib = FALSE</code> is equivalent to <code>isolated = TRUE</code> .
python	Run the embedded-Python validation when its managed <code>.pth</code> file is ready.
quiet	Suppress verification output.

Value

TRUE on success and FALSE on failure.

Index

`gdal_build_gdalraster`, [2](#)
`gdal_build_gdalraster()`, [4](#)
`gdal_enable_python`, [3](#)
`gdal_home`, [4](#)
`gdal_install_runtime`, [4](#)
`gdal_setup`, [5](#)
`gdal_setup()`, [8](#)
`gdal_sitrep`, [6](#)
`gdal_uninstall`, [7](#)
`gdal_update`, [7](#)
`gdal_verify`, [8](#)
`gdal_verify()`, [6](#), [8](#)