

Package ‘geobounds’

September 5, 2026

Title Download Administrative Boundary Data from 'geoBoundaries'

Version 1.0.1

Description Provides tools to download individual country boundaries and global composite boundaries from 'geoBoundaries' [<https://www.geoboundaries.org/>](https://www.geoboundaries.org/) across multiple administrative ('ADM') levels. Returns boundaries as 'sf' objects for mapping and spatial analysis. Runfola et al. (2020) [<doi:10.1371/journal.pone.0231866>](https://doi.org/10.1371/journal.pone.0231866) describe the underlying database.

License MIT + file LICENSE

URL <https://dieghernan.github.io/geobounds/>,
<https://github.com/dieghernan/geobounds>

BugReports <https://github.com/dieghernan/geobounds/issues>

Depends R (>= 4.1.0)

Imports cli, countrycode, dplyr, httr2 (>= 1.0.0), sf, tools, utils

Suggests ggplot2, jsonlite, knitr, quarto, testthat (>= 3.0.0),
tibble, withr

VignetteBuilder quarto

Config/Needs/website dieghernan/gitdevr, xfun, devtools, remotes,
reactable

Config/roxygen2/markdown TRUE

Config/roxygen2/version 8.1.0

Config/testthat/edition 3

Config/testthat/parallel true

Copyright Package code is MIT licensed. See file COPYRIGHTS for data and figure attribution.

Encoding UTF-8

Language en-US

X-schema.org-keywords administrative-boundaries, api, geoboundaries,
r, r-package, spatial-data, cran, cran-r

NeedsCompilation no

Author Diego Hernangómez [aut, cre, cph] (ORCID:
<<https://orcid.org/0000-0001-8457-4658>>),
William & Mary geoLab [dct] (ROR: <<https://ror.org/03hsf0573>>, for the
geoBoundaries project)

Maintainer Diego Hernangómez <diego.hernangomezherrero@gmail.com>

Repository CRAN

Date/Publication 2026-09-05 19:00:02 UTC

Contents

gb_clear_cache	2
gb_detect_cache_dir	3
gb_get	4
gb_get_adm	6
gb_get_max_adm_lvl	10
gb_get_metadata	11
gb_get_world	13
gb_set_cache_dir	15
Index	18

gb_clear_cache	<i>Clear the Rhrefhttps://CRAN.R-project.org/package=geoboundsgeobounds cache directory</i>
----------------	--

Description

Use this function with caution. It clears cached archives and configuration by deleting the **geobounds** configuration directory returned by `tools::R_user_dir()`, deleting the active cache directory and clearing the GEOBOUNDS_CACHE_DIR environment variable. See `base::Sys.getenv()`.

Usage

`gb_clear_cache(config = FALSE, cached_data = TRUE, quiet = TRUE)`

Arguments

- `config` A logical value. If TRUE, delete the **geobounds** configuration directory.
- `cached_data` A logical value. If TRUE, delete the active cache directory and all its contents.
- `quiet` A logical value. If TRUE, suppress informational messages.

Details

This reset restores the cache state of a fresh **geobounds** installation. For safety, the function refuses to recursively delete a cache path that contains the home, working, temporary or package configuration directory. Use a dedicated cache subdirectory. Inspect the active directory with `gb_detect_cache_dir()` before clearing it. Use `gb_set_cache_dir()` to configure a cache directory again.

Value

Invisibly returns NULL. This function is called for its side effects.

See Also

Cache management functions: `gb_detect_cache_dir()`, `gb_set_cache_dir()`

Examples

```
# Caution: this may modify your current state.

## Not run:
my_cache <- gb_detect_cache_dir()
# Set an example cache directory.
ex <- file.path(tempdir(), "example", "cache")
gb_set_cache_dir(ex, quiet = TRUE)

gb_clear_cache(quiet = FALSE)

# Restore the initial cache.
gb_set_cache_dir(my_cache)
identical(my_cache, gb_detect_cache_dir())

## End(Not run)
```

gb_detect_cache_dir	<i>Detect the R</i> hrefhttps://CRAN.R-project.org/package=geobounds geobounds <i>cache directory</i>
---------------------	--

Description

Detects the active cache directory. See `gb_set_cache_dir()`.

Usage

```
gb_detect_cache_dir(x = NULL)
```

Arguments

x	An object. Ignored.
---	---------------------

Value

A character scalar containing the path to the active cache directory. The path is also printed as a clickable message. See [cli::inline-markup](#) from **cli**.

See Also

Cache management functions: [gb_clear_cache\(\)](#), [gb_set_cache_dir\(\)](#)

Examples

```
gb_detect_cache_dir()
```

gb_get

*Download individual country boundaries from **geoBoundaries***

Description

Returns individual country boundaries that reflect how countries represent their own boundaries, without special identification of disputed areas.

Use [gb_get_world\(\)](#) for global composite boundaries that standardize disputed areas and fill gaps between borders.

Boundaries downloaded through this function are not covered by the package's MIT license. Always acknowledge **geoBoundaries** when sharing the boundaries or derived products. See <https://www.geoboundaries.org/index.html#usage>. Consult the boundary metadata for any additional source attribution, license link, share-alike notice or modification statement required by the boundary's license.

The wrappers [gb_get_adm0\(\)](#), [gb_get_adm1\(\)](#), [gb_get_adm2\(\)](#), [gb_get_adm3\(\)](#), [gb_get_adm4\(\)](#) and [gb_get_adm5\(\)](#) are also available for requesting a single ADM level.

Usage

```
gb_get(
  country,
  adm_lvl = "adm0",
  simplified = FALSE,
  release_type = c("gbOpen", "gbHumanitarian", "gbAuthoritative"),
  quiet = TRUE,
  overwrite = FALSE,
  cache_dir = NULL
)
```

Arguments

country	A character vector of country names or ISO 3166-1 alpha-3 country codes. Use "all" to return boundaries for all countries. See also <code>countrycode::countrycode()</code> from countrycode .
adm_lvl	ADM level. Accepted values are "all" (all available boundaries) or the ADM level ("adm0" is the country boundary, "adm1" is the first level of subnational boundaries, "adm2" is the second level and so on). Uppercase versions ("ADM1") and level numbers (0, 1, 2, 3, 4, 5) are also accepted, including numbers supplied as text (for example, "1").
simplified	A logical value. If TRUE, return boundaries that are less accurate but faster to render. The default FALSE uses the primary geoBoundaries layer. See the simplified boundary downloads: https://www.geoboundaries.org/simplifiedDownloads.html .
release_type	A character string, one of "gbOpen", "gbHumanitarian" or "gbAuthoritative". For most users, use "gbOpen" (the default), which contains openly licensed boundaries suitable for most purposes when their individual license terms are followed. "gbHumanitarian" boundaries are mirrored from UN OCHA and may have additional conditions. "gbAuthoritative" boundaries are mirrored from UN SALB , verified through in-country processes and cannot be used for commercial purposes.
quiet	A logical value. If TRUE, suppress informational messages.
overwrite	A logical value. If TRUE, force a fresh download of the source .zip archive.
cache_dir	A path to a cache directory. If not set (the default NULL), boundary archives are stored in the default cache directory (see <code>gb_set_cache_dir()</code>). If no cache directory has been set, archives are stored in a temporary cache directory. See <code>base::tempdir()</code> and the cache strategies in <code>gb_set_cache_dir()</code> .

Details

Each individual country boundary layer is governed by the original license identified in its boundary metadata. See `gb_get_metadata()`. The "gbOpen" release contains multiple open licenses, including ODbL and CC BY-SA. Do not assume every boundary is licensed only under CC BY 4.0. Users should cite the sources listed in the metadata and comply with any attribution, share-alike or non-commercial terms.

Value

An **sf** object from **sf** containing the requested boundaries. Returns NULL if no boundaries match the request or the downloads return no geometries.

Source

geoBoundaries API.

References

Runfola et al. (2020) "geoBoundaries: A global database of political administrative boundaries." *PLOS ONE*, **15**(4), 1–9. doi:[10.1371/journal.pone.0231866](https://doi.org/10.1371/journal.pone.0231866).

See Also

`gb_get_metadata()` inspects boundary metadata and licensing. `gb_get_max_adm_lvl()` checks the ADM levels available for individual country boundaries. `gb_set_cache_dir()` configures where downloaded archives are cached.

Boundary download functions: `gb_get_adm`, `gb_get_world()`

Examples

```
# Map ADM2 in Sri Lanka.
sri_lanka <- gb_get(
  "Sri Lanka",
  adm_lvl = 2,
  simplified = TRUE
)

sri_lanka

library(ggplot2)
ggplot(sri_lanka) +
  geom_sf() +
  labs(
    caption = paste(
      "Sources: geoBoundaries, OpenStreetMap and Wambacher,",
      "license: ODbL 1.0"
    )
  )
)

# Inspect boundary metadata.
library(dplyr)
gb_get_metadata(
  "Sri Lanka",
  adm_lvl = 2
) |>
# Check the individual license.
select(boundaryISO, boundaryType, licenseDetail, licenseSource) |>
glimpse()
```

`gb_get_adm`

Download country boundaries for one ADM level

Description

These functions call `gb_get()` for a single ADM level. `gb_get_adm0()` returns country boundaries, `gb_get_adm1()` returns first-level subnational boundaries (for example, states in the United States) and `gb_get_adm2()` returns second-level subnational boundaries (for example, counties in

the United States). `gb_get_adm3()`, `gb_get_adm4()` and `gb_get_adm5()` return third-, fourth- and fifth-level administrative boundaries, respectively.

Not all countries have the same number of ADM levels. Use `gb_get_max_adm_lvl()` to check availability.

Boundaries downloaded through these functions are not covered by the package's MIT license. Always acknowledge **geoBoundaries** when sharing the boundaries or derived products. See <https://www.geoboundaries.org/index.html#usage>. Consult the boundary metadata for any additional source attribution, license link, share-alike notice or modification statement required by the boundary's license.

Usage

```
gb_get_adm0(  
  country,  
  simplified = FALSE,  
  release_type = c("gbOpen", "gbHumanitarian", "gbAuthoritative"),  
  quiet = TRUE,  
  overwrite = FALSE,  
  cache_dir = NULL  
)  
  
gb_get_adm1(  
  country,  
  simplified = FALSE,  
  release_type = c("gbOpen", "gbHumanitarian", "gbAuthoritative"),  
  quiet = TRUE,  
  overwrite = FALSE,  
  cache_dir = NULL  
)  
  
gb_get_adm2(  
  country,  
  simplified = FALSE,  
  release_type = c("gbOpen", "gbHumanitarian", "gbAuthoritative"),  
  quiet = TRUE,  
  overwrite = FALSE,  
  cache_dir = NULL  
)  
  
gb_get_adm3(  
  country,  
  simplified = FALSE,  
  release_type = c("gbOpen", "gbHumanitarian", "gbAuthoritative"),  
  quiet = TRUE,  
  overwrite = FALSE,  
  cache_dir = NULL  
)
```

```

gb_get_adm4(
  country,
  simplified = FALSE,
  release_type = c("gbOpen", "gbHumanitarian", "gbAuthoritative"),
  quiet = TRUE,
  overwrite = FALSE,
  cache_dir = NULL
)

gb_get_adm5(
  country,
  simplified = FALSE,
  release_type = c("gbOpen", "gbHumanitarian", "gbAuthoritative"),
  quiet = TRUE,
  overwrite = FALSE,
  cache_dir = NULL
)

```

Arguments

country	A character vector of country names or ISO 3166-1 alpha-3 country codes. Use "all" to return boundaries for all countries. See also <code>countrycode::countrycode()</code> from countrycode .
simplified	A logical value. If TRUE, return boundaries that are less accurate but faster to render. The default FALSE uses the primary geoBoundaries layer. See the simplified boundary downloads: https://www.geoboundaries.org/simplifiedDownloads.html .
release_type	A character string, one of "gbOpen", "gbHumanitarian" or "gbAuthoritative". For most users, use "gbOpen" (the default), which contains openly licensed boundaries suitable for most purposes when their individual license terms are followed. "gbHumanitarian" boundaries are mirrored from UN OCHA and may have additional conditions. "gbAuthoritative" boundaries are mirrored from UN SALB , verified through in-country processes and cannot be used for commercial purposes.
quiet	A logical value. If TRUE, suppress informational messages.
overwrite	A logical value. If TRUE, force a fresh download of the source .zip archive.
cache_dir	A path to a cache directory. If not set (the default NULL), boundary archives are stored in the default cache directory (see <code>gb_set_cache_dir()</code>). If no cache directory has been set, archives are stored in a temporary cache directory. See <code>base::tempdir()</code> and the cache strategies in <code>gb_set_cache_dir()</code> .

Details

Each individual country boundary layer is governed by the original license identified in its boundary metadata. See `gb_get_metadata()`. The "gbOpen" release contains multiple open licenses, including ODbL and CC BY-SA. Do not assume every boundary is licensed only under CC BY 4.0. Users should cite the sources listed in the metadata and comply with any attribution, share-alike or non-commercial terms.

Value

An [sf](#) object from [sf](#) containing the requested boundaries. Returns NULL if no boundaries match the request or the downloads return no geometries.

Source

[geoBoundaries API](#).

References

Runfola et al. (2020) "geoBoundaries: A global database of political administrative boundaries." *PLOS ONE*, **15**(4), 1–9. doi:[10.1371/journal.pone.0231866](https://doi.org/10.1371/journal.pone.0231866).

See Also

[gb_get_metadata\(\)](#) inspects boundary metadata and licensing. [gb_get_max_adm_lvl\(\)](#) checks the ADM levels available for individual country boundaries. [gb_set_cache_dir\(\)](#) configures where downloaded archives are cached.

Boundary download functions: [gb_get\(\)](#), [gb_get_world\(\)](#)

Examples

```
lev2 <- gb_get_adm2(
  c("Italia", "Suiza", "Austria"),
  simplified = TRUE
)

library(ggplot2)

ggplot(lev2) +
  geom_sf(aes(fill = shapeGroup)) +
  labs(
    title = "Second-level administrative boundaries",
    subtitle = "Selected countries",
    caption = paste(
      "Sources: geoBoundaries and the original boundary providers,",
      "check gb_get_metadata() for licenses"
    )
  )
```

gb_get_max_adm_lvl *Find the highest available ADM level*

Description

Returns a summary of selected country codes and their highest available ADM level in **geoBoundaries**.

Usage

```
gb_get_max_adm_lvl(
  country = "all",
  release_type = c("gbOpen", "gbHumanitarian", "gbAuthoritative")
)
```

Arguments

country	A character vector of country names or ISO 3166-1 alpha-3 country codes. Use "all" to return boundaries for all countries. See also countrycode::countrycode() from countrycode .
release_type	A character string, one of "gbOpen", "gbHumanitarian" or "gbAuthoritative". For most users, use "gbOpen" (the default), which contains openly licensed boundaries suitable for most purposes when their individual license terms are followed. "gbHumanitarian" boundaries are mirrored from UN OCHA and may have additional conditions. "gbAuthoritative" boundaries are mirrored from UN SALB , verified through in-country processes and cannot be used for commercial purposes.

Value

A [tibble](#) from **tibble** containing ISO 3166-1 alpha-3 country codes and their highest available ADM levels.

Source

geoBoundaries API.

See Also

[gb_get\(\)](#) downloads boundaries for the available ADM levels. The [ADM wrappers](#) request a single administrative level.

Metadata and licensing functions: [gb_get_metadata\(\)](#)

Examples

```
all <- gb_get_max_adm_lvl()
library(dplyr)

# Countries whose highest available level is ADM1.
all |>
  filter(maxBoundaryType == 1)

# Countries with ADM4 available.
all |>
  filter(maxBoundaryType == 4)
```

gb_get_metadata	<i>Retrieve boundary metadata from geoBoundaries</i>
-----------------	---

Description

Returns boundary metadata from the **geoBoundaries API**.

Usage

```
gb_get_metadata(
  country = "all",
  adm_lvl = "all",
  release_type = c("gbOpen", "gbHumanitarian", "gbAuthoritative")
)
```

Arguments

country	A character vector of country names or ISO 3166-1 alpha-3 country codes. Use "all" to return boundaries for all countries. See also countrycode::countrycode() from countrycode .
adm_lvl	ADM level. Accepted values are "all" (all available boundaries) or the ADM level ("adm0" is the country boundary, "adm1" is the first level of subnational boundaries, "adm2" is the second level and so on). Uppercase versions ("ADM1") and level numbers (0, 1, 2, 3, 4, 5) are also accepted, including numbers supplied as text (for example, "1").
release_type	A character string, one of "gbOpen", "gbHumanitarian" or "gbAuthoritative". For most users, use "gbOpen" (the default), which contains openly licensed boundaries suitable for most purposes when their individual license terms are followed. "gbHumanitarian" boundaries are mirrored from UN OCHA and may have additional conditions. "gbAuthoritative" boundaries are mirrored from UN SALB , verified through in-country processes and cannot be used for commercial purposes.

Details

The result is a **tibble** from **tibble** with the following columns:

- **boundaryID**: The ID for this layer. It combines the ISO code, boundary type and a unique identifier generated from the input metadata and geometry. This only changes if the underlying boundary changes.
- **boundaryName**: The name of the country represented by the layer.
- **boundaryISO**: The ISO 3166-1 alpha-3 code for the country.
- **boundaryYearRepresented**: The year or range of years in "START to END" format that the boundary layers represent.
- **boundaryType**: The type of boundary.
- **boundaryCanonical**: The canonical name of the boundary.
- **boundarySource**: A comma-separated list of the primary sources for the boundary.
- **boundarySourceURL**: The URL of the original boundary source.
- **boundaryLicense**: The original license under which the primary source released the boundary.
- **licenseDetail**: Notes about the license.
- **licenseSource**: The URL of the primary source.
- **sourceDataUpdateDate**: The date the source information was integrated into the **geoBoundaries** repository.
- **buildDate**: The date the source boundary was most recently standardized and built into a **geoBoundaries** release.
- **Continent**: The continent the country is associated with.
- **UNSDG-region**: The United Nations Sustainable Development Goals (SDG) region the country is associated with.
- **UNSDG-subregion**: The United Nations Sustainable Development Goals (SDG) subregion the country is associated with.
- **worldBankIncomeGroup**: The World Bank income group the country is associated with.
- **admUnitCount**: The number of administrative units in the boundary.
- **meanVertices**: The mean number of vertices defining the boundaries of each administrative unit in the layer.
- **minVertices**: The minimum number of vertices defining a boundary.
- **maxVertices**: The maximum number of vertices defining a boundary.
- **minPerimeterLengthKM**: The minimum perimeter length of an administrative unit in the layer, measured in kilometers and based on a World Equidistant Cylindrical projection.
- **meanPerimeterLengthKM**: The mean perimeter length of an administrative unit in the layer, measured in kilometers and based on a World Equidistant Cylindrical projection.
- **maxPerimeterLengthKM**: The maximum perimeter length of an administrative unit in the layer, measured in kilometers and based on a World Equidistant Cylindrical projection.
- **meanAreaSqKM**: The mean area of all administrative units in the layer, measured in square kilometers and based on an EASE-GRID 2 projection.

- `minAreaSqKM`: The minimum area of an administrative unit in the layer, measured in square kilometers and based on an EASE-GRID 2 projection.
- `maxAreaSqKM`: The maximum area of an administrative unit in the layer, measured in square kilometers and based on an EASE-GRID 2 projection.
- `staticDownloadLink`: The static download link for the aggregate ZIP file containing all boundary information.
- `gjDownloadURL`: The static download link for the GeoJSON.
- `tjDownloadURL`: The static download link for the TopoJSON.
- `imagePreview`: The static download link for an automatically rendered PNG image of the layer.
- `simplifiedGeometryGeoJSON`: The static download link for the simplified GeoJSON.

Value

A [tibble](#) from [tibble](#) with one row per matching boundary and the columns described in **Details**.

Source

[geoBoundaries API](#).

See Also

[gb_get\(\)](#) downloads the boundaries described by the metadata. The [ADM wrappers](#) request a single administrative level.

Metadata and licensing functions: [gb_get_max_adm_lvl\(\)](#)

Examples

```
# Get boundary metadata for ADM4.

library(dplyr)

gb_get_metadata(adm_lvl = "ADM4") |>
  glimpse()
```

`gb_get_world`

Download global composite boundaries from [geoBoundaries](#)

Description

Returns global composite boundaries for the requested ADM level. Boundaries are clipped to international borders, with gaps between borders filled.

CGAZ boundaries are not covered by the package's MIT license. [Attribution](#) is required when sharing the boundaries or derived products.

Usage

```
gb_get_world(
  country = "all",
  adm_lvl = "adm0",
  quiet = TRUE,
  overwrite = FALSE,
  cache_dir = NULL
)
```

Arguments

country	A character vector of country names or ISO 3166-1 alpha-3 country codes. Use "all" to return boundaries for all countries. See also countrycode::countrycode() from countrycode .
adm_lvl	ADM level. Accepted values are levels 0, 1 and 2 ("adm0" is the country boundary, "adm1" is the first level of subnational boundaries and "adm2" is the second level). Uppercase versions ("ADM1") and level numbers (0, 1, 2) are also accepted, including numbers supplied as text (for example, "1").
quiet	A logical value. If TRUE, suppress informational messages.
overwrite	A logical value. If TRUE, force a fresh download of the source .zip archive.
cache_dir	A path to a cache directory. If not set (the default NULL), boundary archives are stored in the default cache directory (see gb_set_cache_dir()). If no cache directory has been set, archives are stored in a temporary cache directory. See base::tempdir() and the cache strategies in gb_set_cache_dir() .

Details

Comprehensive Global Administrative Zones (CGAZ) are global composites for administrative boundaries. Compared with individual country boundaries, global composite boundaries use extensive simplification so file sizes are small enough for most desktop software. They remove disputed areas, replace them with polygons following United States Department of State definitions and fill gaps between borders.

Follow the citation and use information included in the downloaded CGAZ archive. CGAZ and figures derived from it are not relicensed under the package's MIT license.

Value

An [sf](#) object from **sf** containing the requested boundaries. Returns NULL if no boundaries match the request or the downloads return no geometries.

Source

- **geoBoundaries** global downloads: <https://www.geoboundaries.org/globalDownloads.html>.
- CGAZ release files: <https://github.com/wmgeolab/geoBoundaries/tree/main/releaseData/CGAZ>.

References

Runfola et al. (2020) "geoBoundaries: A global database of political administrative boundaries." *PLOS ONE*, **15**(4), 1–9. doi:[10.1371/journal.pone.0231866](https://doi.org/10.1371/journal.pone.0231866).

See Also

[gb_get_metadata\(\)](#) inspects boundary metadata and licensing. [gb_get_max_adm_lvl\(\)](#) checks the ADM levels available for individual country boundaries. [gb_set_cache_dir\(\)](#) configures where downloaded archives are cached.

Boundary download functions: [gb_get\(\)](#), [gb_get_adm](#)

Examples

```
# This download may take some time.
## Not run:
world <- gb_get_world()

library(ggplot2)

ggplot(world) +
  geom_sf() +
  coord_sf(expand = FALSE) +
  labs(caption = "Source: geoBoundaries (CGAZ)")

## End(Not run)
```

gb_set_cache_dir	<i>Set the <code>Rhref</code>https://CRAN.R-project.org/package=geobounds cache directory</i>
------------------	--

Description

Sets the active cache directory and optionally saves it for future sessions. Use [gb_detect_cache_dir\(\)](#) to find the active cache directory.

Usage

```
gb_set_cache_dir(
  cache_dir = NULL,
  overwrite = FALSE,
  install = FALSE,
  quiet = FALSE
)
```

Arguments

cache_dir	A path to a cache directory. If NULL, the function stores cached archives in a temporary directory. See base::tempdir() .
overwrite	A logical value. If TRUE, replace a cache directory already saved in the configuration file.
install	A logical value. If TRUE, save the cache directory for use in future sessions. Defaults to FALSE. If cache_dir is NULL, this parameter is set to FALSE automatically.
quiet	A logical value. If TRUE, suppress informational messages.

Details

By default, when no cache_dir is set, **geobounds** uses a directory inside [base::tempdir\(\)](#). Cached archives in this directory are removed when the R session ends. To reuse a cache directory across R sessions, use `gb_set_cache_dir(cache_dir = "a/path/here", install = TRUE)`. This saves the directory in the user configuration path returned by [tools::R_user_dir\(\)](#).

Value

An invisible character scalar containing the path to the cache directory.

Cache strategies

- For occasional use, use the default temporary cache directory.
- Set the cache directory for the current session with `gb_set_cache_dir(cache_dir = "a/path/here")`.
- Save a persistent cache directory for future R sessions with `gb_set_cache_dir(cache_dir = "a/path/here", install = TRUE)`.
- Set the cache directory for an individual download with the cache_dir argument. See [gb_get\(\)](#).

See Also

[gb_get\(\)](#) and [gb_get_world\(\)](#) use the configured cache directory for downloaded boundary archives. [tools::R_user_dir\(\)](#) identifies standard locations for user-specific files.

Cache management functions: [gb_clear_cache\(\)](#), [gb_detect_cache_dir\(\)](#)

Examples

```
# Caution: this may modify your current state.

## Not run:
my_cache <- gb_detect_cache_dir()

# Set an example cache directory.
ex <- file.path(tempdir(), "example", "cachenew")
gb_set_cache_dir(ex)

gb_detect_cache_dir()
```

```
# Restore the initial cache.  
gb_set_cache_dir(my_cache)  
identical(my_cache, gb_detect_cache_dir())  
  
## End(Not run)  
  
gb_detect_cache_dir()
```

Index

- * **api**
 - gb_get, 4
 - gb_get_adm, 6
 - gb_get_world, 13
- * **cache**
 - gb_clear_cache, 2
 - gb_detect_cache_dir, 3
 - gb_set_cache_dir, 15
- * **metadata**
 - gb_get_max_adm_lvl, 10
 - gb_get_metadata, 11

ADM wrappers, 10, 13

base::Sys.getenv(), 2

base::tempdir(), 5, 8, 14, 16

cli::inline-markup, 4

countrycode::countrycode(), 5, 8, 10, 11, 14

gb_clear_cache, 2

gb_clear_cache(), 4, 16

gb_detect_cache_dir, 3

gb_detect_cache_dir(), 3, 15, 16

gb_get, 4

gb_get(), 6, 9, 10, 13, 15, 16

gb_get_adm, 6, 6, 15

gb_get_adm0 (gb_get_adm), 6

gb_get_adm0(), 4, 6

gb_get_adm1 (gb_get_adm), 6

gb_get_adm1(), 4, 6

gb_get_adm2 (gb_get_adm), 6

gb_get_adm2(), 4, 6

gb_get_adm3 (gb_get_adm), 6

gb_get_adm3(), 4, 7

gb_get_adm4 (gb_get_adm), 6

gb_get_adm4(), 4, 7

gb_get_adm5 (gb_get_adm), 6

gb_get_adm5(), 4, 7

gb_get_cgaz (gb_get_world), 13

gb_get_max_adm_lvl, 10

gb_get_max_adm_lvl(), 6, 7, 9, 13, 15

gb_get_meta (gb_get_metadata), 11

gb_get_metadata, 11

gb_get_metadata(), 5, 6, 8–10, 15

gb_get_world, 13

gb_get_world(), 4, 6, 9, 16

gb_set_cache_dir, 15

gb_set_cache_dir(), 3–6, 8, 9, 14, 15

sf, 5, 9, 14

tibble, 10, 12, 13

tools::R_user_dir(), 2, 16