

# Package ‘rdataretriever’

September 17, 2020

**Title** R Interface to the Data Retriever

**Description** Provides an R interface to the Data Retriever <<https://retriever.readthedocs.io/en/latest/>> via the Data Retriever's command line interface. The Data Retriever automates the tasks of finding, downloading, and cleaning public datasets, and then stores them in a local database.

**Version** 3.0.0

**BugReports** <https://github.com/ropensci/rdataretriever/issues>

**URL** <https://docs.ropensci.org/rdataretriever/> (website),  
<https://github.com/ropensci/rdataretriever/>

**Depends** R (>= 3.4.0)

**Imports** reticulate (>= 1.16), semver

**Suggests** testthat (>= 1.0.0), DBI, devtools, RSQLite, RPostgreSQL

**SystemRequirements** Python (>= 3.0), retriever (>= 3.0.0) (version must be listed to patch to allow parsing)

**License** MIT + file LICENSE

**LazyData** true

**RoxygenNote** 7.1.1

**NeedsCompilation** no

**Author** Daniel McGlinn [aut] (<<https://orcid.org/0000-0003-2359-3526>>),  
Pranita Sharma [aut] (<<https://orcid.org/0000-0002-5871-380X>>),  
David J Harris [aut] (<<https://orcid.org/0000-0003-3332-9307>>),  
Henry Senyondo [aut, cre] (<<https://orcid.org/0000-0001-7105-5808>>),  
Hao Ye [aut] (<<https://orcid.org/0000-0002-8630-1458>>),  
Shawn Taylor [aut] (<<https://orcid.org/0000-0002-6178-6903>>),  
Apoorva Pandey [aut] (<<https://orcid.org/0000-0001-9834-4415>>),  
Harshit Bansal [aut] (<<https://orcid.org/0000-0002-3285-812X>>),  
Max Pohlman [aut],  
Ethan White [aut] (<<https://orcid.org/0000-0001-6728-7745>>)

**Maintainer** Henry Senyondo <[henrykironde@gmail.com](mailto:henrykironde@gmail.com)>

**Repository** CRAN

**Date/Publication** 2020-09-17 09:40:03 UTC

**R topics documented:**

check_for_updates . . . . .	2
check_retriever_availability . . . . .	3
commit . . . . .	3
commit_log . . . . .	4
datasets . . . . .	4
dataset_names . . . . .	5
data_retriever_version . . . . .	5
download . . . . .	6
fetch . . . . .	7
get_citation . . . . .	7
get_dataset_names_upstream . . . . .	8
get_script_citation . . . . .	8
get_script_upstream . . . . .	9
get_updates . . . . .	9
install . . . . .	10
install_csv . . . . .	11
install_json . . . . .	12
install_msaccess . . . . .	13
install_mysql . . . . .	14
install_postgres . . . . .	15
install_retriever . . . . .	16
install_sqlite . . . . .	16
install_xml . . . . .	17
reload_scripts . . . . .	18
reset . . . . .	18
use_RetrieverPath . . . . .	19
<b>Index</b>	<b>20</b>

---

check_for_updates	<i>Check for updates</i>
-------------------	--------------------------

---

**Description**

Check for updates

**Usage**

```
check_for_updates(repo = "")
```

**Arguments**

repo            path to the repository

**Examples**

```
rdataretriever::check_for_updates()
```

---

```
check_retriever_availability
```

*Check to see if minimum version of retriever Python package is installed*

---

**Description**

Check to see if minimum version of retriever Python package is installed

**Usage**

```
check_retriever_availability()
```

**Value**

boolean

**Examples**

```
rdataretriever::check_retriever_availability()
```

---

```
commit
```

*Commit a dataset*

---

**Description**

Commit a dataset

**Usage**

```
commit(dataset, commit_message = "", path = NULL, quiet = FALSE)
```

**Arguments**

dataset	name of the dataset
commit_message	commit message for the commit
path	path to save the committed dataset, if no path given save in provenance directory
quiet	logical, if true retriever runs in quiet mode

**Examples**

```
rdataretriever::commit("iris")
```

---

commit_log	<i>See the log of committed dataset stored in provenance directory</i>
------------	--

---

**Description**

See the log of committed dataset stored in provenance directory

**Usage**

```
commit_log(dataset)
```

**Arguments**

dataset	name of the dataset stored in provenance directory
---------	--

**Examples**

```
rdataretriever::commit_log("iris")
```

---

datasets	<i>Name all available dataset scripts.</i>
----------	--

---

**Description**

Additional information on the available datasets can be found at url <https://retriever.readthedocs.io/en/latest/datasets.html>

**Usage**

```
datasets(keywords = "", licenses = "")
```

**Arguments**

keywords	Search all datasets by keywords
licenses	Search all datasets by licenses

**Value**

returns a character vector with the available datasets for download

**Examples**

```
rdataretriever::datasets()
```

---

dataset_names	<i>Name all available dataset scripts.</i>
---------------	--

---

**Description**

Additional information on the available datasets can be found at url <https://retriever.readthedocs.io/en/latest/datasets.html>

**Usage**

```
dataset_names()
```

**Value**

returns a character vector with the available datasets for download

**Examples**

```
rdataretriever::dataset_names()
```

---

data_retriever_version	<i>Get Data Retriever version</i>
------------------------	-----------------------------------

---

**Description**

Get Data Retriever version

**Usage**

```
data_retriever_version(clean = TRUE)
```

**Arguments**

clean            boolean return cleaned version appropriate for semver

**Value**

returns a string with the version information

## Examples

```
rdataretriever::data_retriever_version()
```

---

download

*Download datasets via the Data Retriever.*

---

## Description

Directly downloads data files with no processing, allowing downloading of non-tabular data.

## Usage

```
download(  
  dataset,  
  path = "./",  
  quiet = FALSE,  
  sub_dir = "",  
  debug = FALSE,  
  use_cache = TRUE  
)
```

## Arguments

dataset	the name of the dataset that you wish to download
path	the path where the data should be downloaded to
quiet	logical, if true retriever runs in quiet mode
sub_dir	downloaded dataset is stored into a custom subdirectory.
debug	Setting TRUE helps in debugging in case of errors
use_cache	Setting FALSE reinstalls scripts even if they are already installed

## Examples

```
rdataretriever::download("plant-comp-ok")  
# downloaded files will be copied to your working directory  
# when no path is specified  
dir()
```

---

fetch	<i>Fetch a dataset via the Data Retriever</i>
-------	---

---

### Description

Each datafile in a given dataset is downloaded to a temporary directory and then imported as a data.frame as a member of a named list.

### Usage

```
fetch(dataset, quiet = TRUE, data_names = NULL)
```

### Arguments

dataset	the names of the dataset that you wish to download
quiet	logical, if true retriever runs in quiet mode
data_names	the names you wish to assign to cells of the list which stores the fetched dataframes. This is only relevant if you are downloading more than one dataset.

### Examples

```
## fetch the portal Database
portal <- rdataretriever::fetch("portal")
class(portal)
names(portal)
## preview the data in the portal species datafile
head(portal$species)
vegdata <- rdataretriever::fetch(c("plant-comp-ok", "plant-occur-oosting"))
names(vegdata)
names(vegdata$plant_comp_ok)
```

---

get_citation	<i>Get dataset citation information and a description</i>
--------------	---

---

### Description

Get dataset citation information and a description

### Usage

```
get_citation(dataset)
```

### Arguments

dataset	name of the dataset
---------	---------------------

**Value**

returns a string with the citation information

**Examples**

```
rdataretriever::get_citation("plant-comp-us")
```

---

```
get_dataset_names_upstream
```

*Get dataset names from upstream*

---

**Description**

Get dataset names from upstream

**Usage**

```
get_dataset_names_upstream(keywords = "", licenses = "", repo = "")
```

**Arguments**

keywords	filter datasets based on keywords
licenses	filter datasets based on license
repo	path to the repository

**Examples**

```
rdataretriever::get_dataset_names_upstream(keywords = "", licenses = "", repo = "")
```

---

```
get_script_citation
```

*Get citation*

---

**Description**

Get citation

**Usage**

```
get_script_citation(dataset = "")
```



**Arguments**

dataset            dataset to obtain citation

**Examples**

```
rdataretriever::get_script_citation(dataset = "")
```

---

*get\_script\_upstream*    *Get script upstream*

---

**Description**

Get script upstream

**Usage**

```
get_script_upstream(dataset, repo = "")
```

**Arguments**

dataset            name of the dataset  
repo                path to the repository

**Examples**

```
rdataretriever::get_script_upstream("iris")
```

---

*get\_updates*            *Update the retriever's dataset scripts to the most recent versions.*

---

**Description**

This function will check if the version of the retriever's scripts in your local directory '`~/retriever/scripts/`' is up-to-date with the most recent official retriever release. Note it is possible that even more updated scripts exist at the retriever repository <https://github.com/weecology/retriever/tree/master/scripts> that have not yet been incorporated into an official release, and you should consider checking that page if you have any concerns.

**Usage**

```
get_updates()
```

**Examples**

```
rdataretriever::get_updates()
```

---

```
install
```

---

*Install datasets via the Data Retriever (deprecated).*

---

**Description**

Data is stored in either CSV files or one of the following database management systems: MySQL, PostgreSQL, SQLite, or Microsoft Access.

**Usage**

```
install(
  dataset,
  connection,
  db_file = NULL,
  conn_file = NULL,
  data_dir = ".",
  log_dir = NULL
)
```

**Arguments**

dataset	the name of the dataset that you wish to download
connection	what type of database connection should be used. The options include: mysql, postgres, sqlite, msaccess, or csv'
db_file	the name of the database file the dataset should be loaded into
conn_file	the path to the .conn file that contains the connection configuration options for mysql and postgres databases. This defaults to mysql.conn or postgres.conn respectively. The connection file is a file that is formatted in the following way: <pre>           host      my_server@my_host.com           port      my_port_number           user       my_user_name           password   my_password       </pre>
data_dir	the location where the dataset should be installed. Only relevant for csv connection types. Defaults to current working directory
log_dir	the location where the retriever log should be stored if the progress is not printed to the console

## Examples

```
rdataretriever::install("iris", "csv")
```

---

install\_csv

*Install datasets via the Data Retriever.*

---

## Description

Data is stored in CSV files

## Usage

```
install_csv(  
  dataset,  
  table_name = "{db}_{table}.csv",  
  data_dir = getwd(),  
  debug = FALSE,  
  use_cache = TRUE,  
  force = FALSE,  
  hash_value = NULL  
)
```

## Arguments

dataset	the name of the dataset that you wish to install or path to a committed dataset zip file
table_name	the name of the database file to store data
data_dir	the dir path to store data, defaults to working dir
debug	Setting TRUE helps in debugging in case of errors
use_cache	Setting FALSE reinstalls scripts even if they are already installed
force	Setting TRUE doesn't prompt for confirmation while installing committed datasets when changes are discovered in environment
hash_value	the hash value of committed dataset when installing from provenance directory

## Examples

```
rdataretriever::install_csv("iris")
```

---

`install_json`*Install datasets via the Data Retriever.*

---

### Description

Data is stored in JSON files

### Usage

```
install_json(  
  dataset,  
  table_name = "{db}_{table}.json",  
  data_dir = getwd(),  
  debug = FALSE,  
  use_cache = TRUE,  
  force = FALSE,  
  hash_value = NULL  
)
```

### Arguments

<code>dataset</code>	the name of the dataset that you wish to install or path to a committed dataset zip file
<code>table_name</code>	the name of the database file to store data
<code>data_dir</code>	the dir path to store data, defaults to working dir
<code>debug</code>	Setting TRUE helps in debugging in case of errors
<code>use_cache</code>	Setting FALSE reinstalls scripts even if they are already installed
<code>force</code>	Setting TRUE doesn't prompt for confirmation while installing committed datasets when changes are discovered in environment
<code>hash_value</code>	the hash value of committed dataset when installing from provenance directory

### Examples

```
rdataretriever::install_json("iris")
```

---

install_msaccess	<i>Install datasets via the Data Retriever.</i>
------------------	---

---

### Description

Data is stored in MSAccess database

### Usage

```
install_msaccess(  
  dataset,  
  file = "access.mdb",  
  table_name = "[{db} {table}]",  
  debug = FALSE,  
  use_cache = TRUE,  
  force = FALSE,  
  hash_value = NULL  
)
```

### Arguments

dataset	the name of the dataset that you wish to install or path to a committed dataset zip file
file	file name for database
table_name	table name for installing of dataset
debug	Setting TRUE helps in debugging in case of errors
use_cache	Setting FALSE reinstalls scripts even if they are already installed
force	Setting TRUE doesn't prompt for confirmation while installing committed datasets when changes are discovered in environment
hash_value	the hash value of committed dataset when installing from provenance directory

### Examples

```
rdataretriever::install_msaccess(dataset = "iris", file = "sqlite.db")
```

---

install_mysql	<i>Install datasets via the Data Retriever.</i>
---------------	---

---

### Description

Data is stored in MySQL database

### Usage

```
install_mysql(
  dataset,
  user = "root",
  password = "",
  host = "localhost",
  port = 3306,
  database_name = "{db}",
  table_name = "{db}.{table}",
  debug = FALSE,
  use_cache = TRUE,
  force = FALSE,
  hash_value = NULL
)
```

### Arguments

dataset	the name of the dataset that you wish to install or path to a committed dataset zip file
user	username for database connection
password	password for database connection
host	hostname for connection
port	port number for connection
database_name	database name in which dataset will be installed
table_name	table name specified especially for datasets containing one file
debug	Setting TRUE helps in debugging in case of errors
use_cache	Setting FALSE reinstalls scripts even if they are already installed
force	Setting TRUE doesn't prompt for confirmation while installing committed datasets when changes are discovered in environment
hash_value	the hash value of committed dataset when installing from provenance directory

### Examples

```
rdataretriever::install_mysql(dataset = "portal", user = "postgres", password = "abcdef")
```

---

install_postgres	<i>Install datasets via the Data Retriever.</i>
------------------	---

---

### Description

Data is stored in PostgreSQL database

### Usage

```
install_postgres(
  dataset,
  user = "postgres",
  password = "",
  host = "localhost",
  port = 5432,
  database = "postgres",
  database_name = "{db}",
  table_name = "{db}.{table}",
  bbox = list(),
  debug = FALSE,
  use_cache = TRUE,
  force = FALSE,
  hash_value = NULL
)
```

### Arguments

dataset	the name of the dataset that you wish to install or path to a committed dataset zip file
user	username for database connection
password	password for database connection
host	hostname for connection
port	port number for connection
database	the database name default is postgres
database_name	database schema name in which dataset will be installed
table_name	table name specified especially for datasets containing one file
bbox	Optional extent values used to fetch data from the spatial dataset
debug	Setting TRUE helps in debugging in case of errors
use_cache	Setting FALSE reinstalls scripts even if they are already installed
force	Setting TRUE doesn't prompt for confirmation while installing committed datasets when changes are discovered in environment
hash_value	the hash value of committed dataset when installing from provenance directory

**Examples**

```
rdataretriever::install_postgres(dataset = "portal", user = "postgres", password = "abcdef")
```

---

```
install_retriever      install the python module 'retriever'
```

---

**Description**

install the python module 'retriever'

**Usage**

```
install_retriever(method = "auto", conda = "auto")
```

**Arguments**

method	Installation method. By default, "auto" automatically finds a method that will work in the local environment. Change the default to force a specific installation method. Note that the "virtualenv" method is not available on Windows.
conda	The path to a conda executable. Use "auto" to allow reticulate to automatically find an appropriate conda binary. See <b>Finding Conda</b> for more details.

---

```
install_sqlite      Install datasets via the Data Retriever.
```

---

**Description**

Data is stored in SQLite database

**Usage**

```
install_sqlite(
  dataset,
  file = "sqlite.db",
  table_name = "{db}_{table}",
  data_dir = getwd(),
  debug = FALSE,
  use_cache = TRUE,
  force = FALSE,
  hash_value = NULL
)
```



**Arguments**

dataset	the name of the dataset that you wish to install or path to a committed dataset zip file
file	Sqlite database file name or path
table_name	table name for installing of dataset
data_dir	the dir path to store the db, defaults to working dir
debug	Setting TRUE helps in debugging in case of errors
use_cache	Setting FALSE reinstalls scripts even if they are already installed
force	Setting TRUE doesn't prompt for confirmation while installing committed datasets when changes are discovered in environment
hash_value	the hash value of committed dataset when installing from provenance directory

**Examples**

```
rdataretriever::install_sqlite(dataset = "iris", file = "sqlite.db")
```

---

install\_xml

---

*Install datasets via the Data Retriever.*


---

**Description**

Data is stored in XML files

**Usage**

```
install_xml(
  dataset,
  table_name = "{db}_{table}.xml",
  data_dir = getwd(),
  debug = FALSE,
  use_cache = TRUE,
  force = FALSE,
  hash_value = NULL
)
```

**Arguments**

dataset	the name of the dataset that you wish to install or path to a committed dataset zip file
table_name	the name of the database file to store data
data_dir	the dir path to store data, defaults to working dir
debug	Setting TRUE helps in debugging in case of errors

use_cache	Setting FALSE reinstalls scripts even if they are already installed
force	Setting TRUE doesn't prompt for confirmation while installing committed datasets when changes are discovered in environment
hash_value	the hash value of committed dataset when installing from provenance directory

### Examples

```
rdataretriever::install_xml("iris")
```

---

reload_scripts	<i>Update the retriever's global_script_list with the scripts present in the ~/.retriever directory.</i>
----------------	--

---

### Description

Update the retriever's global\_script\_list with the scripts present in the ~/.retriever directory.

### Usage

```
reload_scripts()
```

### Examples

```
rdataretriever::reload_scripts()
```

---

reset	<i>Reset the scripts or data(raw_data) directory or both</i>
-------	--

---

### Description

Reset the scripts or data(raw\_data) directory or both

### Usage

```
reset(scope = "all")
```

### Arguments

scope	All resets both scripst and data directory
-------	--

### Examples

```
rdataretriever::reset("iris")
```

---

use\_RetrieverPath      *Setting path of retriever*

---

**Description**

Setting path of retriever

**Usage**

use\_RetrieverPath(path)

**Arguments**

path                      location of retriever in the system

**Examples**

```
rdataretriever::use_RetrieverPath("/home/<system_name>/anaconda2/envs/py27/bin/")
```

# Index

## \* **utilities**

- get\_updates, 9
  
- check\_for\_updates, 2
- check\_retriever\_availability, 3
- commit, 3
- commit\_log, 4
  
- data\_retriever\_version, 5
- dataset\_names, 5
- datasets, 4
- download, 6
  
- fetch, 7
  
- get\_citation, 7
- get\_dataset\_names\_upstream, 8
- get\_script\_citation, 8
- get\_script\_upstream, 9
- get\_updates, 9
  
- install, 10
- install\_csv, 11
- install\_json, 12
- install\_msaccess, 13
- install\_mysql, 14
- install\_postgres, 15
- install\_retriever, 16
- install\_sqlite, 16
- install\_xml, 17
  
- reload\_scripts, 18
- reset, 18
  
- use\_RetrieverPath, 19