Package ‘arcpy’

May 7, 2026

Title Interface to 'ArcGIS' 'Python' Modules

Version 0.4-0

Description An interface to the 'ArcGIS' 'arcpy' and 'arcgis' 'python' API <https://pro.arcgis.com/en/pro-app/latest/arcpy/get-started/arcgis-api-for-python.htm>.
Provides various tools for installing and configuring a 'Conda' environment for accessing 'ArcGIS' geoprocessing functions. Helper functions for manipulating and converting 'ArcGIS' objects from R are also provided.

URL https://github.com/mkoohafkan/arcpy,

https://hydroecology.net/arcpy/

BugReports https://github.com/mkoohafkan/arcpy/issues

SystemRequirements Conda (>= 4.6), ArcGIS Pro (>= 3.0)

Depends R (>= 4.2)

Imports stats, reticulate (>= 1.31)

Suggests tibble (>= 3.0), knitr (>= 1.21), rmarkdown (>= 1.11), testthat (>= 2.1.0), rstudioapi (>= 0.7), sf, terra

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.2.3

VignetteBuilder knitr

Config/testthat/edition 3

NeedsCompilation no

Author Michael Koohafkan [aut, cre]

Maintainer Michael Koohafkan <michael.koohafkan@gmail.com>

Repository CRAN

Date/Publication 2024-01-22 17:32:51 UTC

Contents

arcpy-package 2
arcpy_version 2
da_drop 3
da_fields 4
da_insert 4
da_read 5
da_update 6
install_arcpy 7
to_arcpy 8

Index **9**

`arcpy-package` *Interface to ArcPy*

Description

An interface to the ArcGIS `arcpy` Python module via the R-Python interface provided by `reticulate`. Loading the packages exposes the `arcpy` and `arcgis` python modules for accessing the ArcGIS geoprocessor. See the vignettes to get started.

Author(s)

Maintainer: Michael Koohafkan <`michael.koohafkan@gmail.com`>

See Also

`install_arcpy()`

`arcpy_version` *Get Arcpy Version*

Description

Verify the supplied arcpy module version and identify the required Python version.

Usage

```
arcpy_version(version, conda = "auto", channel = "esri", forge = TRUE)
```

Arguments

`version`	The arcpy module version.
`conda`	The path to a conda executable. Use `"auto"` to allow `reticulate` to automatically find an appropriate conda binary. See **Finding Conda** and `conda_binary()` for more details.
`channel`	An optional character vector of conda channels to include. When specified, the `forge` argument is ignored. If you need to specify multiple channels, including the conda forge, you can use `c("conda-forge", <other channels>)`.
`forge`	Boolean; include the conda-forge repository?

Value

A named list providing version numbers of arcpy and Python.

---

da_drop — *Table Row Removal with arcpy.da*

Description

Drop records from a table (e.g. attribute table of a layer) with the arcpy.da module.

Usage

```
da_drop(table.path, rows)
```

Arguments

`table.path`	The file path to the table.
`rows`	The row indexes to drop.

Value

(Invisible) The path to the table, i.e. `table.path`.

Examples

```
## Not run:
arcpy$env$workspace = tempdir()
arcpy$env$scratchWorkspace = tempdir()
fc = arcpy$management$CopyFeatures(system.file("CA_Counties",
  "CA_Counties_TIGER2016.shp", package = "arcpy"), "CA_Counties")
d = da_read(fc, c("STATEFP", "COUNTYFP"))
drop.rows = which(d$STATEFP == "06" & d$COUNTYFP == "067")
da_drop(fc, drop.rows)

## End(Not run)
```

da_fields — *List Attribute Table Fields*

Description

Read attribute table field names with ‘arcpy.da“ module.

Usage

```
da_fields(table.path)
```

Arguments

`table.path`	The file path to the table.

Value

A vector of field names.

Examples

```
## Not run:
arcpy$env$workspace = tempdir()
arcpy$env$scratchWorkspace = tempdir()
fc = arcpy$management$CopyFeatures(system.file("CA_Counties",
  "CA_Counties_TIGER2016.shp", package = "arcpy"), "CA_Counties")
da_fields(fc)

## End(Not run)
```

da_insert — *Table Insertion with arcpy.da*

Description

Insert records into a table (e.g. attribute table of a layer) with the arcpy.da module.

Usage

```
da_insert(table.path, d)
```

Arguments

`table.path`	The file path to the table.
`d`	The data to write to `table.path`, with the same number of rows as the table. Column names must match field names of the table.

Value

(Invisible) The path to the table, i.e. `table.path`.

Examples

```
## Not run:
arcpy$env$workspace = tempdir()
arcpy$env$scratchWorkspace = tempdir()
fc = arcpy$management$CopyFeatures(system.file("CA_Counties",
  "CA_Counties_TIGER2016.shp", package = "arcpy"), "CA_Counties")
d = da_read(fc, c("ALAND", "CLASSFP"))
add.d = data.frame(ALAND= 1e4, CLASSFP = "H2",
  stringsAsFactors = FALSE)
da_insert(fc, add.d)

## End(Not run)
```

---

da_read — *Read Table with arcpy.da*

Description

Read a table (e.g. attribute table of a layer) with the `arcpy.da` module.

Usage

```
da_read(table.path, fields, simplify = TRUE)
```

Arguments

`table.path`	The file path to the table.
`fields`	A vector of field names or column indices to retrieve.
`simplify`	If `TRUE`, coerce the results to a data.frame. If `FALSE`, the results will be returned as a list of lists, with each top-level element corresponding to one row of the table.

Details

This implementation may be faster than accessing the `@data` slot of an object created from `rgdal::readOGR` in cases where there are a very large number of features. An additional advantage of `da_read` is that it can read raster attribute tables and stand-alone tables stored in file geodatabases, which is not supported by `rgdal::readOGR`.

Value

a dataframe with columns corresponding to `fields`.

Examples

```
## Not run:
arcpy$env$workspace = tempdir()
arcpy$env$scratchWorkspace = tempdir()
fc = arcpy$management$CopyFeatures(system.file("CA_Counties",
  "CA_Counties_TIGER2016.shp", package = "arcpy"), "CA_Counties")
da_read(fc, c("COUNTYFP", "ALAND"))

## End(Not run)
```

---

da_update *Update Table with arcpy.da*

Description

Update a table (e.g., attribute table of a layer) with the arcpy.da module.

Usage

```
da_update(table.path, d)
```

Arguments

table.path	The file path to the table.
d	The data to write to table.path, with the same number of rows as the table. Column names must match field names of the table.

Value

(Invisible) The path to the table, i.e. table.path.

Examples

```
## Not run:
arcpy$env$workspace = tempdir()
arcpy$env$scratchWorkspace = tempdir()
fc = arcpy$management$CopyFeatures(system.file("CA_Counties",
  "CA_Counties_TIGER2016.shp", package = "arcpy"), "CA_Counties")
d = da_read(fc, "ALAND")
d["ALAND"] = d$ALAND+ 5000
da_update(fc, d)

## End(Not run)
```

install_arcpy *Install ArcGIS Conda Environment*

Description

Create a Conda environment with the "arcpy" module.

Usage

```
install_arcpy(
  method = "conda",
  conda = "auto",
  version = NULL,
  envname = "r-arcpy",
  extra_packages = NULL,
  restart_session = TRUE,
  python_version = NULL,
  channel = "esri",
  forge = TRUE,
  ...,
  new_env = identical(envname, "r-arcpy")
)
```

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 **Finding Conda** and conda_binary() for more details.
version	Arcpy version to install. Note that the requested arcpy version must match your ArcGIS Pro version.
envname	The name, or full path, of the environment in which Python packages are to be installed. When NULL (the default), the active environment as set by the RETICULATE_PYTHON_ENV variable will be used; if that is unset, then the r-reticulate environment will be used.
extra_packages	Additional Python packages to install along with arcpy.
restart_session	Restart R session after installing (note this will only occur within RStudio).
python_version	Pass a string like "3.9" to request that conda install a specific Python version. Note that the Python version must be compatible with the requested arcpy version. If NULL, the latest compatible Python version will be used.
channel	An optional character vector of conda channels to include. When specified, the forge argument is ignored. If you need to specify multiple channels, including the conda forge, you can use c("conda-forge", <other channels>).

forge	Boolean; include the conda-forge repository?
...	other arguments passed to `reticulate::conda_install()`.
new_env	If TRUE, any existing Python conda environment specified by envname is deleted first.

Details

The Conda environment must be configured to match the ArcGIS Pro version currently installed. If ArcGIS Pro is updated, the Conda environment must be recreated.

Value

(Invisible) TRUE if the Conda environment was created successfully.

`to_arcpy` *R to/from ArcGIS Object*

Description

Convert rasters and features between R and ArcGIS.

Usage

```
to_arcpy(x, ...)

from_arcpy(x, ...)
```

Arguments

x	A raster or feature.
...	Reservered for future expansion.

Value

For `to_arcpy()`, an ArcGIS raster or feature layer. For `from_arcpy()`, a `SpatRaster` or `sf` object.

Index

_PACKAGE *(arcpy-package)*, 2

arcpy *(arcpy-package)*, 2
arcpy-package, 2
arcpy_version, 2

conda_binary(), *3*, *7*

da_drop, 3
da_fields, 4
da_insert, 4
da_read, 5
da_update, 6

from_arcpy *(to_arcpy)*, 8

install_arcpy, 7
install_arcpy(), *2*

reticulate::conda_install(), *8*

to_arcpy, 8