

Package: connector.sharepoint (via r-universe)

June 1, 2026

Title 'Microsoft SharePoint' Interface for the 'connector' Package

Version 0.1.0

Description Extends the 'connector' package to provide a convenient interface for accessing and interacting with 'Microsoft SharePoint' directly from 'R'. Supports listing, reading, writing, uploading, downloading, and removing files and directories on 'SharePoint' document libraries. Authentication is handled through 'Azure' tokens via the 'AzureAuth' package.

License Apache License (>= 2)

URL <https://novonordisk-opensource.github.io/connector.sharepoint/>,
<https://github.com/novonordisk-opensource/connector.sharepoint>

BugReports <https://github.com/NovoNordisk-OpenSource/connector.sharepoint/issues>

Depends R (>= 4.1.0)

Imports AzureAuth, checkmate, cli, connector (>= 1.0.0),
Microsoft365R, R6, tools, zephyr, rlang

Suggests knitr, rmarkdown, testthat (>= 3.0.0), tibble, withr, whirl
(>= 0.2.0), glue, mockery

VignetteBuilder knitr, rmarkdown

Config/Needs/website rmarkdown

Config/testthat/edition 3

Encoding UTF-8

RoxygenNote 7.3.3

NeedsCompilation no

Author Vladimir Obucina [aut, cre] (ORCID:
<<https://orcid.org/0009-0008-2608-2661>>), Cervan Girard [aut],
Aksel Thomsen [aut], Steffen Falgreen Larsen [aut], Novo
Nordisk A/S [cph]

Maintainer Vladimir Obucina <vlob@novonordisk.com>

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

Date/Publication 2026-06-01 08:50:06 UTC

RemoteUrl <https://github.com/cran/connector.sharepoint>

RemoteRef HEAD

RemoteSha c1ae1bb5c37980e193bf727d73725949fbee8d84

Contents

connector-options-sharepoint	2
connector_sharepoint	3
ConnectorSharepoint	4
create_directory_cnt	6
disconnect_cnt	7
download_cnt	7
download_directory_cnt	8
get_default_hash	9
get_token	9
list_content_cnt	10
log_read_connector.ConnectorSharepoint	10
log_remove_connector.ConnectorSharepoint	11
log_write_connector.ConnectorSharepoint	12
read_cnt	13
remove_cnt	14
remove_directory_cnt	15
tbl_cnt	15
upload_cnt	16
upload_directory_cnt	17
write_cnt	18
Index	20

connector-options-sharepoint
Options for connector.sharepoint

Description

Configuration options for the connector.sharepoint

overwrite:

Overwrite existing content if it exists in the connector?

- Default: FALSE
- Option: `connector.sharepoint.overwrite`
- Environment: `R_CONNECTOR.SHAREPOINT_OVERWRITE`

verbosity_level:

Verbosity level for functions in connector. See [zephyr::verbosity_level](#) for details.

- Default: "verbose"

- Option: `connector.sharepoint.verbosity_level`
- Environment: `R_CONNECTOR.SHAREPOINT_VERBOSITY_LEVEL`

`connector_sharepoint` *Create sharepoint connector object*

Description

Create a new sharepoint connector object. See [ConnectorSharepoint](#) for details.

Usage

```
connector_sharepoint(  
  site_url,  
  token = get_token(),  
  folder = "",  
  extra_class = NULL,  
  ...  
)
```

Arguments

<code>site_url</code>	character The URL of the Sharepoint site
<code>token</code>	AzureAuth::AzureToken The Azure token. By default, it will be retrieved by get_token
<code>folder</code>	character The path of the folder to interact with, if you don't want to interact with the root folder.
<code>extra_class</code>	character Extra class added to the object.
<code>...</code>	Additional parameters to pass to the ConnectorSharepoint object

Details

The `extra_class` parameter allows you to create a subclass of the [ConnectorSharepoint](#) object. This can be useful if you want to create a custom connection object for easier dispatch of new s3 methods, while still inheriting the methods from the [ConnectorSharepoint](#) object.

Authentication is handled through Azure tokens. See [get_token\(\)](#) for details on token acquisition and management.

Value

A new [ConnectorSharepoint](#) object

Examples

```
## Not run:
# Connect
cs <- connector_sharepoint(Sys.getenv("SHAREPOINT_SITE_URL"))
cs

# Create subclass connection
cs_subclass <- connector_sharepoint(Sys.getenv("SHAREPOINT_SITE_URL"),
  extra_class = "subclass"
)

cs_subclass
class(cs_subclass)

## End(Not run)
```

ConnectorSharepoint *Connector Object for Sharepoint class, built on top of [connector::Connector](#) class*

Description

This object is used to interact with Sharepoint, adding the ability to list, read, write, download, upload, create directories and remove files.

Details

About the token, you can retrieve it by following the guideline in your enterprise.

Super classes

[connector::Connector](#) -> [connector::ConnectorFS](#) -> ConnectorSharepoint

Active bindings

folder [character](#) The path of the folder to interact with

token [character](#) The Azure token

site_url [character](#) The URL of the Sharepoint site

path [character](#) The whole URL path of the Sharepoint resource

Methods**Public methods:**

- [ConnectorSharepoint\\$new\(\)](#)
- [ConnectorSharepoint\\$get_conn\(\)](#)

Method new(): Initializes the ConnectorSharepoint class

Usage:

```
ConnectorSharepoint$new(
  site_url,
  token = get_token(),
  folder = "",
  extra_class = NULL,
  ...
)
```

Arguments:

site_url **character** The URL of the Sharepoint site

token **character** The Azure token. By default, it will be retrieve by `get_token`

folder **character** The path of the folder to interact with, if you don't want to interact with the root folder "Documents"

extra_class **character** Extra class added to the object.

... Additional parameters to pass to the `get_sharepoint_site` function

Returns: A `ConnectorSharepoint` object

Method get_conn(): Get the connection

Usage:

```
ConnectorSharepoint$get_conn()
```

Returns: The connection

Examples

```
## Not run:
# Connect to Sharepoint
cs <- ConnectorSharepoint$new(
  site_url = Sys.getenv("SHAREPOINT_SITE_URL")
)

cs

# List content
cs$list_content_cnt()

# Write to the connector
cs$write_cnt(iris, "iris.rds")

# Check it is there
cs$list_content_cnt()

# Read the result back
cs$read_cnt("iris.rds") |>
  head()

# Remove a file or directory
cs$remove_cnt("iris.rds")
```

```
# Check it is there
cs$list_content_cnt()

## End(Not run)
```

create_directory_cnt *Create a directory*

Description

Addition list content methods for sharepoint connectors implemented for `connector::create_directory_cnt()`:

- `ConnectorSharepoint`: Reuses the `connector::create_directory_cnt()` method for `ConnectorSharepoint()`

Usage

```
create_directory_cnt(connector_object, name, open = TRUE, ...)
```

```
## S3 method for class 'ConnectorSharepoint'
create_directory_cnt(connector_object, name, open = TRUE, ...)
```

Arguments

connector_object	<code>Connector</code> The connector object to use.
name	<code>character</code> The name of the directory to create
open	create a new connector object
...	<code>ConnectorSharepoint</code> : Additional parameters to pass to <code>create_folder()</code> method of <code>ms_drive</code> class.

Value

`ConnectorSharepoint` object or `ConnectorSharepoint` object of a newly built directory

disconnect_cnt	<i>Disconnect (close) the connection of the connector</i>
----------------	-----------------------------------------------------------

Description

Generic implementing of how to disconnect from the relevant connections. Mostly relevant for DBI connectors.

- **ConnectorDBI**: Uses `DBI::dbDisconnect()` to create a table reference to close a DBI connection.

Usage

```
disconnect_cnt(connector_object, ...)
```

Arguments

connector_object

Connector The connector object to use.

...

Additional arguments passed to the method for the individual connector.

Value

invisible connector_object.

download_cnt	<i>Download content from the connector</i>
--------------	--------------------------------------------

Description

Addition list content methods for sharepoint connectors implemented for `connector::download_cnt()`:

- **ConnectorSharepoint**: Reuses the `connector::download_cnt()` method for `ConnectorSharepoint()`

Usage

```
download_cnt(connector_object, src, dest = basename(src), ...)
```

```
## S3 method for class 'ConnectorSharepoint'
```

```
download_cnt(connector_object, src, dest = basename(src), ...)
```

Arguments

connector_object	Connector The connector object to use.
src	The name of the file to download from SharePoint
dest	The local path to save the downloaded file
...	ConnectorSharepoint : Additional parameters to pass to the download() method of ms_drive class.

Value

[ConnectorSharepoint](#) object

download_directory_cnt
Download a directory

Description

Addition list content methods for sharepoint connectors implemented for `connector::download_directory_cnt()`:

Usage

```
download_directory_cnt(connector_object, src, dest = basename(src), ...)
```

```
## S3 method for class 'ConnectorSharepoint'
download_directory_cnt(
  connector_object,
  src,
  dest = basename(src),
  ...,
  recursive = TRUE
)
```

Arguments

connector_object	Connector The connector object to use.
src	The name of the directory to download from SharePoint
dest	The local directory path to save the downloaded content
...	additional parameters passed on to download_folder() method of ms_drive class.
recursive	If recursive is TRUE, all subfolders will also be transferred recursively. Default: FALSE

Value

[ConnectorSharepoint](#) object

get_default_hash	<i>Get the default hash</i>
------------------	-----------------------------

Description

Two solutions are possible:

1. The user has set the environment variable SHAREPOINT_AZURE_HASH
2. The user has set the file .active_hash in the AzureR directory

Usage

```
get_default_hash()
```

Value

A character string with the token hash, or NULL if not found

Examples

```
## Not run:  
hash <- get_default_hash()  
  
## End(Not run)
```

get_token	<i>Retrieve a token from the AzureAuth package</i>
-----------	----------------------------------------------------

Description

Retrieve a token from the AzureAuth package

Usage

```
get_token(hash = get_default_hash())
```

Arguments

hash	The hash of the token to use. By default, use this function to retrieve it get_default_hash . If not found, use the first token found.
------	--------------------------------------------------------------------------------------------------------------------------------------------------------

Value

An [AzureAuth::AzureToken](#) object

Examples

```
## Not run:
token <- get_token()

## End(Not run)
```

list_content_cnt	<i>List available content from the connector</i>
------------------	--------------------------------------------------

Description

Addition list content methods for sharepoint connectors implemented for `connector::list_content_cnt()`:

- **ConnectorSharepoint**: Reuses the `connector::list_content_cnt()` method for `ConnectorSharepoint()`

Usage

```
list_content_cnt(connector_object, ...)

## S3 method for class 'ConnectorSharepoint'
list_content_cnt(connector_object, ...)
```

Arguments

`connector_object` **Connector** The connector object to use.

... **ConnectorSharepoint**: Additional parameters to pass to the `list_items()` method of `ms_drive` class.

Value

A **character** vector of content names

log_read_connector.ConnectorSharepoint	<i>Connector Logging Functions</i>
----------------------------------------	------------------------------------

Description

- **ConnectorSharepoint**: Implementation of the `log_read_connector` function for the `ConnectorSharepoint` class.

Addition log read methods for sharepoint connectors implemented for `connector::log_read_connector()`:

Usage

```
## S3 method for class 'ConnectorSharepoint'
log_read_connector(connector_object, name, ...)

log_read_connector(connector_object, name, ...)
```

Arguments

connector_object	The connector object to log operations for. Can be any connector class (ConnectorFS, ConnectorDBI, ConnectorLogger, etc.)
name	Character string specifying the name or identifier of the resource being operated on (e.g., file name, table name)
...	Additional parameters passed to specific method implementations. May include connector-specific options or metadata.

Details**Connector Logging Functions**

The logging system is built around S3 generic functions that dispatch to specific implementations based on the connector class. Each operation is logged with contextual information including connector details, operation type, and resource names.

Value

These are primarily side-effect functions that perform logging. The actual return value depends on the specific method implementation, typically:

- log_read_connector: Result of the read operation
- log_write_connector: Invisible result of write operation
- log_remove_connector: Invisible result of remove operation
- log_list_content_connector: List of connector contents

```
log_remove_connector.ConnectorSharepoint
      Connector Logging Functions
```

Description

- **ConnectorSharepoint**: Implementation of the log_remove_connector function for the ConnectorSharepoint class.

Addition log remove methods for sharepoint connectors implemented for `connector::log_remove_connector()`:

Usage

```
## S3 method for class 'ConnectorSharepoint'
log_remove_connector(connector_object, name, ...)

log_remove_connector(connector_object, name, ...)
```

Arguments

connector_object	The connector object to log operations for. Can be any connector class (ConnectorFS, ConnectorDBI, ConnectorLogger, etc.)
name	Character string specifying the name or identifier of the resource being operated on (e.g., file name, table name)
...	Additional parameters passed to specific method implementations. May include connector-specific options or metadata.

Details**Connector Logging Functions**

The logging system is built around S3 generic functions that dispatch to specific implementations based on the connector class. Each operation is logged with contextual information including connector details, operation type, and resource names.

Value

These are primarily side-effect functions that perform logging. The actual return value depends on the specific method implementation, typically:

- log_read_connector: Result of the read operation
- log_write_connector: Invisible result of write operation
- log_remove_connector: Invisible result of remove operation
- log_list_content_connector: List of connector contents

```
log_write_connector.ConnectorSharepoint
      Connector Logging Functions
```

Description

- [ConnectorSharepoint](#): Implementation of the log_write_connector function for the ConnectorSharepoint class.

Addition log write methods for sharepoint connectors implemented for `connector::log_write_connector()`:

Usage

```
## S3 method for class 'ConnectorSharepoint'
log_write_connector(connector_object, name, ...)

log_write_connector(connector_object, name, ...)
```

Arguments

connector_object	The connector object to log operations for. Can be any connector class (ConnectorFS, ConnectorDBI, ConnectorLogger, etc.)
name	Character string specifying the name or identifier of the resource being operated on (e.g., file name, table name)
...	Additional parameters passed to specific method implementations. May include connector-specific options or metadata.

Details**Connector Logging Functions**

The logging system is built around S3 generic functions that dispatch to specific implementations based on the connector class. Each operation is logged with contextual information including connector details, operation type, and resource names.

Value

These are primarily side-effect functions that perform logging. The actual return value depends on the specific method implementation, typically:

- `log_read_connector`: Result of the read operation
- `log_write_connector`: Invisible result of write operation
- `log_remove_connector`: Invisible result of remove operation
- `log_list_content_connector`: List of connector contents

read_cnt	<i>Read content from the connector</i>
----------	----------------------------------------

Description

Addition read methods for sharepoint connectors implemented for `connector::read_cnt()`:

- **ConnectorSharepoint**: Reuses the `connector::read_cnt()` method for `ConnectorSharepoint()`

Usage

```
read_cnt(connector_object, name, ...)

## S3 method for class 'ConnectorSharepoint'
read_cnt(connector_object, name, ...)
```

Arguments

connector_object [Connector](#) The connector object to use.

name The name of the file to read

... [ConnectorSharepoint](#): Additional parameters to pass to the `get_item()` method of `ms_drive` class

Value

R object with the content. For rectangular data a [data.frame](#).

remove_cnt	<i>Remove content from the connector</i>
------------	------------------------------------------

Description

Addition remove methods for sharepoint connectors implemented for `connector::remove_cnt()`:

- [ConnectorSharepoint](#): Reuses the `connector::remove_cnt()` method for `ConnectorSharepoint()`

Usage

```
remove_cnt(connector_object, name, ...)

## S3 method for class 'ConnectorSharepoint'
remove_cnt(connector_object, name, ...)
```

Arguments

connector_object [Connector](#) The connector object to use.

name [character](#) Name of the content to read, write, or remove. Typically the table name.

... [ConnectorSharepoint](#): Additional parameters to pass to the `delete()` method of `ms_drive` class.

Value

[ConnectorSharepoint](#) object

remove_directory_cnt *Remove a directory*

Description

Addition list content methods for sharepoint connectors implemented for `connector::remove_directory_cnt()`:

- **ConnectorSharepoint**: Reuses the `connector::remove_directory_cnt()` method for `ConnectorSharepoint()`

Usage

```
remove_directory_cnt(connector_object, name, ...)
```

```
## S3 method for class 'ConnectorSharepoint'
remove_directory_cnt(connector_object, name, ...)
```

Arguments

connector_object

Connector The connector object to use.

name **character** The name of the directory to remove

... **ConnectorSharepoint**: Additional parameters to pass to `delete()` method of `ms_drive` class.

Value

ConnectorSharepoint object

tbl_cnt *Use dplyr verbs to interact with the remote database table*

Description

Addition tbl methods for sharepoint connectors implemented for `connector::tbl_cnt()`:

- **ConnectorSharepoint**: Uses `read_cnt()` to allow redundancy.

Usage

```
tbl_cnt(connector_object, name, ...)
```

```
## S3 method for class 'ConnectorSharepoint'
tbl_cnt(connector_object, name, ...)
```

Arguments

connector_object	Connector The connector object to use.
name	character Name of the content to read, write, or remove. Typically the table name.
...	Additional arguments passed to the method for the individual connector.

Value

A [dplyr::tbl](#) object.

upload_cnt	<i>Upload content to the connector</i>
------------	----------------------------------------

Description

Addition list content methods for sharepoint connectors implemented for [connector::upload_cnt\(\)](#):

- [ConnectorSharepoint](#): Reuses the [connector::upload_cnt\(\)](#) method for [ConnectorSharepoint\(\)](#)

Usage

```
upload_cnt(
  connector_object,
  src,
  dest = basename(src),
  overwrite = zephyr::get_option("overwrite", "connector"),
  ...
)

## S3 method for class 'ConnectorSharepoint'
upload_cnt(
  connector_object,
  src,
  dest = basename(src),
  overwrite = zephyr::get_option("overwrite", "connector.sharepoint"),
  ...,
  recursive = FALSE
)
```

Arguments

connector_object	Connector The connector object to use.
src	The local file path to upload
dest	The destination name/path in SharePoint

overwrite	Overwrite existing content if it exists in the connector? See connector-options for details. Default can be set globally with <code>options(connector.overwrite = TRUE/FALSE)</code> or environment variable <code>R_CONNECTOR_OVERWRITE..</code> Default: FALSE.
...	ConnectorSharepoint : Additional parameters to pass to the <code>upload_file()</code> or <code>upload_folder()</code> method of ms_drive class.
recursive	If recursive is TRUE, all subfolders will also be transferred recursively. Default: FALSE

Value

[ConnectorSharepoint](#) object

upload_directory_cnt *Upload a directory*

Description

Addition list content methods for sharepoint connectors implemented for [connector::upload_directory_cnt\(\)](#):

Usage

```
upload_directory_cnt(
  connector_object,
  src,
  dest,
  overwrite = zephyr::get_option("overwrite", "connector"),
  open = FALSE,
  ...
)

## S3 method for class 'ConnectorSharepoint'
upload_directory_cnt(
  connector_object,
  src,
  dest = basename(src),
  overwrite = zephyr::get_option("overwrite", "connector.sharepoint"),
  open = FALSE,
  ...,
  recursive = TRUE
)
```

Arguments

`connector_object` [Connector](#) The connector object to use.

`src` The local directory path to upload

dest	The destination directory name/path in SharePoint
overwrite	Overwrite existing content if it exists in the connector? See connector-options for details. Default can be set globally with <code>options(connector.overwrite = TRUE/FALSE)</code> or environment variable <code>R_CONNECTOR_OVERWRITE..</code> Default: FALSE.
open	logical Open the directory as a new connector object.
...	additional parameters passed on to <code>upload_folder()</code> method of ms_drive class.
recursive	If recursive is TRUE, all subfolders will also be transferred recursively. Default: FALSE

Value

[ConnectorSharepoint](#) object

write_cnt	<i>Write content to the connector</i>
-----------	---------------------------------------

Description

Addition write methods for sharepoint connectors implemented for `connector::write_cnt()`:

- [ConnectorSharepoint](#): Reuses the `connector::write_cnt()` method for [ConnectorSharepoint\(\)](#)

Usage

```
write_cnt(
  connector_object,
  x,
  name,
  overwrite = zephyr::get_option("overwrite", "connector"),
  ...
)

## S3 method for class 'ConnectorSharepoint'
write_cnt(
  connector_object,
  x,
  name,
  overwrite = zephyr::get_option("overwrite", "connector.sharepoint"),
  ...
)
```

Arguments

connector_object	Connector The connector object to use.
x	The object to write to the connection
name	character Name of the content to read, write, or remove. Typically the table name.
overwrite	Overwrite existing content if it exists in the connector? See connector-options for details. Default can be set globally with <code>options(connector.overwrite = TRUE/FALSE)</code> or environment variable <code>R_CONNECTOR_OVERWRITE..</code> Default: <code>FALSE</code> .
...	ConnectorSharepoint : Additional parameters to pass to the <code>connector::write_file()</code> function

Value

[invisible](#) connector_object.

Index

AzureAuth::AzureToken, [3](#), [9](#)

character, [3–6](#), [10](#), [14–16](#), [19](#)
Connector, [6–8](#), [10](#), [14–17](#), [19](#)
connector-options, [17–19](#)
connector-options-sharepoint, [2](#)
connector::Connector, [4](#)
connector::ConnectorFS, [4](#)
connector::create_directory_cnt(), [6](#)
connector::download_cnt(), [7](#)
connector::download_directory_cnt(), [8](#)
connector::list_content_cnt(), [10](#)
connector::log_read_connector(), [10](#)
connector::log_remove_connector(), [11](#)
connector::log_write_connector(), [12](#)
connector::read_cnt(), [13](#)
connector::remove_cnt(), [14](#)
connector::remove_directory_cnt(), [15](#)
connector::tbl_cnt(), [15](#)
connector::upload_cnt(), [16](#)
connector::upload_directory_cnt(), [17](#)
connector::write_cnt(), [18](#)
connector::write_file(), [19](#)
connector_sharepoint, [3](#)
ConnectorDBI, [7](#)
ConnectorSharepoint, [3](#), [4](#), [5–8](#), [10–19](#)
ConnectorSharepoint(), [6](#), [7](#), [10](#), [13–16](#), [18](#)
create_directory_cnt, [6](#)

data.frame, [14](#)
DBI::dbDisconnect(), [7](#)
disconnect_cnt, [7](#)
download_cnt, [7](#)
download_directory_cnt, [8](#)
dplyr::tbl, [16](#)

get_default_hash, [9](#), [9](#)
get_sharepoint_site, [5](#)
get_token, [3](#), [5](#), [9](#)
get_token(), [3](#)

invisible, [7](#), [19](#)

list_content_cnt, [10](#)
log_read_connector
 (log_read_connector.ConnectorSharepoint),
 [10](#)
log_read_connector.ConnectorSharepoint,
 [10](#)
log_remove_connector
 (log_remove_connector.ConnectorSharepoint),
 [11](#)
log_remove_connector.ConnectorSharepoint,
 [11](#)
log_write_connector
 (log_write_connector.ConnectorSharepoint),
 [12](#)
log_write_connector.ConnectorSharepoint,
 [12](#)
logical, [18](#)

ms_drive, [6](#), [8](#), [10](#), [14](#), [15](#), [17](#), [18](#)

read_cnt, [13](#)
remove_cnt, [14](#)
remove_directory_cnt, [15](#)

tbl_cnt, [15](#)

upload_cnt, [16](#)
upload_directory_cnt, [17](#)

write_cnt, [18](#)

zephyr::verbosity_level, [2](#)