

# Package: datr (via r-universe)

February 21, 2025

**Title** 'Dat' Protocol Interface

**Version** 0.1.0.900

**Description** Interface with the 'Dat' p2p network protocol  
<<https://datproject.org>>. Clone archives from the network,  
share your own files, and install packages from the network.

**License** GPL-3

**Depends** R (>= 3.1)

**Suggests** covr, testthat

**SystemRequirements** dat (>= 13.10.0)

**ByteCompile** true

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 6.0.1.9000

**URL** <https://github.com/libscie/datr>

**BugReports** <https://github.com/libscie/datr/issues>

**Repository** <https://libscie.r-universe.dev>

**RemoteUrl** <https://github.com/libscie/datr>

**RemoteRef** HEAD

**RemoteSha** cbaa963ae7910790445691a5dd0b50482a07af70

## Contents

clone_dat . . . . .	2
cmd_check . . . . .	2
create_dat . . . . .	3
datr . . . . .	3
dat_install . . . . .	3
dat_version . . . . .	4
install_dat . . . . .	5
log_dat . . . . .	5

pull_dat . . . . .	6
share_dat . . . . .	6
status_dat . . . . .	7
sync_dat . . . . .	7

<b>Index</b>	<b>8</b>
--------------	----------

---

clone_dat	<i>Clone a Dat link to directory</i>
-----------	--------------------------------------

---

**Description**

Function to clone a Dat link to the machine.

**Usage**

clone\_dat(link, dir)

**Arguments**

- link            Dat link (always clones latest version now)
- dir            Directory to clone to. If non-existent, will create.

**Examples**

clone\_dat(link = 'dat://datr-chris.hashbase.io', dir = tempdir())

---

cmd_check	<i>Checking the command to be passed to Dat system call</i>
-----------	---

---

**Description**

Checking the command to be passed to Dat system call

**Usage**

cmd\_check(x)

**Arguments**

- x              Command to check

---

create_dat	Create Dat metadata
------------	---------------------

---

**Description**

Function to create a dat.json interactively, with title and description.

**Usage**

```
create_dat(dir = ".")
```

**Arguments**

dir	Directory to create Dat. Defaults to working directory.
-----	---

**Examples**

```
create_dat(dir = tempdir())
```

---

datr	<i>datr: Dat Protocol Interface</i>
------	-------------------------------------

---

**Description**

Interface with the 'Dat' p2p network protocol <<https://datproject.org>>. Clone archives from the network, share your own files, and install packages from the network.

---

dat_install	<i>Dat install</i>
-------------	--------------------

---

**Description**

Download the released binaries for Dat and add them to your environment. If a more recent version has been released, you can specify it as argument (we try to update it accordingly, but might sometimes have time off!).

**Usage**

```
dat_install(os = "win", path, ver = "13.10.0")
```

**Arguments**

os	'win' (Windows), 'mac', or 'linux'
path	Where to install Dat
ver	Version. See <a href="https://github.com/datproject/dat/releases">https://github.com/datproject/dat/releases</a>

**Value**

NULL. Prints stdout of terminal along the way.

**Note**

The system 'PATH' will be modified. On Linux-ish systems, ~/.profile will also be modified.

**Examples**

```
## Not run:  
dat_install(os = 'win')  
dat_install(os = 'macos')  
dat_install(os = 'linux')  
  
## End(Not run)
```

---

dat\_version

*Which version of Dat is installed?*

---

**Description**

Which version of Dat is installed?

**Usage**

```
dat_version()
```

**Value**

Dat version string (if installed)

**Examples**

```
dat_version()
```

---

install_dat	<i>Install R package from Dat link</i>
-------------	--

---

**Description**

Directly install an R package from a Dat link.

**Usage**

```
install_dat(link, dir)
```

**Arguments**

link	Dat link
dir	Temporary directory to store download in.

**Value**

Console log.

**Examples**

```
## Not run:  
install_dat('dat://datr-chris.hashbase.io', dir = tempdir())  
  
## End(Not run)
```

---

log_dat	<i>Show Dat archive log</i>
---------	-----------------------------

---

**Description**

Retrieve log of the Dat folder. You can either retrieve the log of a Dat link (argument `link`) or of a local Dat folder (argument `dir`). Defaults to giving the log of the working directory. If a Dat link is provided, this overrides the `dir` argument.

**Usage**

```
log_dat(path)
```

**Arguments**

path	Dat archive, can both be local or on the network.
------	---

**Value**

Console log.

**Examples**

```
log_dat(path = '.')  
log_dat(path = 'dat://pastedat-taravancil.hashbase.io')
```

---

pull\_dat

*Update Dat folder*

---

**Description**

Function to update working directory if it is a Dat folder. This function only receives updates.

**Usage**

```
pull_dat()
```

**Examples**

```
pull_dat()
```

---

share\_dat

*Share folder to network*

---

**Description**

Share directory to the Dat network, until aborted. Requires write access to Dat archive.

**Usage**

```
share_dat(dir = ".")
```

**Arguments**

dir                      Directory to share. Defaults to working directory.

**Examples**

```
share_dat(dir = tempdir())
```

---

status_dat	<i>Status of the Dat folder</i>
------------	---------------------------------

---

**Description**

Get some information Dat folder in the working directory. Includes: Dat link, the number of files, and the current version.

**Usage**

```
status_dat()
```

**Value**

Console log.

**Examples**

```
## Not run:  
status_dat()  
  
## End(Not run)
```

---

sync_dat	<i>Synchronize with Dat network</i>
----------	-------------------------------------

---

**Description**

Synchronize directory with the Dat network. If write access to the Dat is available, this is equivalent to [share\\_dat](#). If read-only access is available, this is equivalent to [pull\\_dat](#).

**Usage**

```
sync_dat()
```

**Examples**

```
sync_dat()
```

# Index

clone\_dat, [2](#)  
cmd\_check, [2](#)  
create\_dat, [3](#)  
  
dat\_install, [3](#)  
dat\_version, [4](#)  
datr, [3](#)  
datr-package (datr), [3](#)  
  
install\_dat, [5](#)  
  
log\_dat, [5](#)  
  
pull\_dat, [6](#), [7](#)  
  
share\_dat, [6](#), [7](#)  
status\_dat, [7](#)  
sync\_dat, [7](#)