



dryxDropboxCL Documentation

Release 0.1.2

Dave Young

2016

1	Installation	3
1.1	Development	3
1.2	Issues	3
2	Usage	5
3	Documentation	7
4	Tutorial	9
5	Installation	11
5.1	Development	11
5.2	Issues	11
6	Usage	13
7	Documentation	15
8	Tutorial	17
8.1	Subpackages	17
8.1.1	dryxDropboxCL (<i>subpackage</i>)	17
8.1.2	dryxDropboxCL.commonutils (<i>subpackage</i>)	17
8.1.3	dryxDropboxCL.original (<i>subpackage</i>)	17
8.2	Modules	17
8.2.1	dryxDropboxCL.cl_utils (<i>module</i>)	17
8.2.2	dryxDropboxCL.utKit (<i>module</i>)	18
8.2.3	dryxDropboxCL.original.dropbox (<i>module</i>)	18
8.3	Classes	19
8.3.1	dryxDropboxCL.utKit.utKit (<i>class</i>)	19
	Methods	19
8.3.2	dryxDropboxCL.original.dropbox.CommandTicker (<i>class</i>)	19
	Methods	20
	Attributes	20
8.3.3	dryxDropboxCL.original.dropbox.DownloadState (<i>class</i>)	20
	Methods	20
8.3.4	dryxDropboxCL.original.dropbox.DropboxCommand (<i>class</i>)	20
	Methods	20
8.3.5	dryxDropboxCL.original.dropbox.SignatureVerifyError (<i>function</i>)	21
8.4	Functions	21

8.4.1	<code>dryxDropboxCL.setup_selective_sync</code> (<i>function</i>)	21
8.4.2	<code>dryxDropboxCL.start_if_not_running</code> (<i>function</i>)	22
8.4.3	<code>dryxDropboxCL.original.dropbox.alias</code> (<i>function</i>)	22
8.4.4	<code>dryxDropboxCL.original.dropbox.autostart</code> (<i>function</i>)	22
8.4.5	<code>dryxDropboxCL.original.dropbox.can_reroll_autostart</code> (<i>function</i>)	22
8.4.6	<code>dryxDropboxCL.original.dropbox.columnize</code> (<i>function</i>)	22
8.4.7	<code>dryxDropboxCL.original.dropbox.command</code> (<i>function</i>)	23
8.4.8	<code>dryxDropboxCL.original.dropbox.console_flush</code> (<i>function</i>)	23
8.4.9	<code>dryxDropboxCL.original.dropbox.console_print</code> (<i>function</i>)	23
8.4.10	<code>dryxDropboxCL.original.dropbox.download</code> (<i>function</i>)	23
8.4.11	<code>dryxDropboxCL.original.dropbox.download_file_chunk</code> (<i>function</i>)	23
8.4.12	<code>dryxDropboxCL.original.dropbox.exclude</code> (<i>function</i>)	23
8.4.13	<code>dryxDropboxCL.original.dropbox.filestatus</code> (<i>function</i>)	23
8.4.14	<code>dryxDropboxCL.original.dropbox.gpgme_context</code> (<i>function</i>)	23
8.4.15	<code>dryxDropboxCL.original.dropbox.grab_link_url_if_necessary</code> (<i>function</i>)	24
8.4.16	<code>dryxDropboxCL.original.dropbox.help</code> (<i>function</i>)	24
8.4.17	<code>dryxDropboxCL.original.dropbox.is_dropbox_running</code> (<i>function</i>)	24
8.4.18	<code>dryxDropboxCL.original.dropbox.lansync</code> (<i>function</i>)	24
8.4.19	<code>dryxDropboxCL.original.dropbox.load_serialized_images</code> (<i>function</i>)	24
8.4.20	<code>dryxDropboxCL.original.dropbox.ls</code> (<i>function</i>)	24
8.4.21	<code>dryxDropboxCL.original.dropbox.methodcaller</code> (<i>function</i>)	24
8.4.22	<code>dryxDropboxCL.original.dropbox.plat</code> (<i>function</i>)	24
8.4.23	<code>dryxDropboxCL.original.dropbox.puburl</code> (<i>function</i>)	24
8.4.24	<code>dryxDropboxCL.original.dropbox.relp</code> (<i>function</i>)	25
8.4.25	<code>dryxDropboxCL.original.dropbox.requires_dropbox_running</code> (<i>function</i>)	25
8.4.26	<code>dryxDropboxCL.original.dropbox.reroll_autostart</code> (<i>function</i>)	25
8.4.27	<code>dryxDropboxCL.original.dropbox.running</code> (<i>function</i>)	25
8.4.28	<code>dryxDropboxCL.original.dropbox.start</code> (<i>function</i>)	25
8.4.29	<code>dryxDropboxCL.original.dropbox.start_dropbox</code> (<i>function</i>)	25
8.4.30	<code>dryxDropboxCL.original.dropbox.status</code> (<i>function</i>)	25
8.4.31	<code>dryxDropboxCL.original.dropbox.stop</code> (<i>function</i>)	25
8.4.32	<code>dryxDropboxCL.original.dropbox.unicode_abspath</code> (<i>function</i>)	25
8.4.33	<code>dryxDropboxCL.original.dropbox.usage</code> (<i>function</i>)	26
8.4.34	<code>dryxDropboxCL.original.dropbox.verify_signature</code> (<i>function</i>)	26
8.4.35	<code>dryxDropboxCL.original.dropbox.yes_no_question</code> (<i>function</i>)	26
8.5	Indexes	26
8.6	Todo	26

Python Module Index

27

Extra helper tools for the dropbox command-line.

Installation

The easiest way to install dryxDropboxCL is to use `pip`:

```
pip install dryxDropboxCL
```

Or you can clone the [github repo](#) and install from a local version of the code:

```
git clone git@github.com:thespacedoctor/dryxDropboxCL.git
cd dryxDropboxCL
python setup.py install
```

To upgrade to the latest version of dryxDropboxCL use the command:

```
pip install dryxDropboxCL --upgrade
```

1.1 Development

If you want to tinker with the code, then install in development mode. This means you can modify the code from your cloned repo:

```
git clone git@github.com:thespacedoctor/dryxDropboxCL.git
cd dryxDropboxCL
python setup.py develop
```

[Pull requests](#) are welcomed!

1.2 Issues

Please report any issues [here](#).

Usage

```
dryxDropboxCL bump
dryxDropboxCL resync -s <pathToSettingsFile>

-h, --help          show this help message
-v, --version        show version
-s, --settings       the settings file
```

Documentation

Documentation for dryxDropboxCL is hosted by [Read the Docs](#) (last stable version and latest version).

Tutorial

Todo

- add tutorial
-

Installation

The easiest way to install dryxDropboxCL is to use `pip`:

```
pip install dryxDropboxCL
```

Or you can clone the [github repo](#) and install from a local version of the code:

```
git clone git@github.com:thespacedoctor/dryxDropboxCL.git
cd dryxDropboxCL
python setup.py install
```

To upgrade to the latest version of dryxDropboxCL use the command:

```
pip install dryxDropboxCL --upgrade
```

5.1 Development

If you want to tinker with the code, then install in development mode. This means you can modify the code from your cloned repo:

```
git clone git@github.com:thespacedoctor/dryxDropboxCL.git
cd dryxDropboxCL
python setup.py develop
```

[Pull requests](#) are welcomed!

5.2 Issues

Please report any issues [here](#).

Usage

```
dryxDropboxCL bump
dryxDropboxCL resync -s <pathToSettingsFile>

-h, --help          show this help message
-v, --version        show version
-s, --settings       the settings file
```

Documentation

Documentation for dryxDropboxCL is hosted by [Read the Docs](#) (last stable version and latest version).

Tutorial

Todo

- add tutorial
-

8.1 Subpackages

<i>dryxDropboxCL</i>
<i>dryxDropboxCL.commonutils</i>
<i>dryxDropboxCL.original</i>

8.1.1 dryxDropboxCL (*subpackage*)

8.1.2 dryxDropboxCL.commonutils (*subpackage*)

8.1.3 dryxDropboxCL.original (*subpackage*)

8.2 Modules

<i>dryxDropboxCL.cl_utils</i>	<i>The CL utils for dryxDropboxCL</i>
<i>dryxDropboxCL.utKit</i>	
<i>dryxDropboxCL.original.dropbox</i>	

8.2.1 dryxDropboxCL.cl_utils (*module*)

The CL utils for dryxDropboxCL

Author David Young

Date Created September 5, 2014

Usage: dryxDropboxCL bump dryxDropboxCL resync -s <pathToSettingsFile>

-h, --help show this help message

-v, --version show version
-s, --settings the settings file

`dryxDropboxCL.cl_utils.main` (*arguments=None*)

The main function used when “cl_utils.py” is run as a single script from the cl, or when installed as a cl command

8.2.2 dryxDropboxCL.utKit (*module*)

class `dryxDropboxCL.utKit.utKit` (*moduleDirectory*)

Override dryx utKit

8.2.3 dryxDropboxCL.original.dropbox (*module*)

`dryxDropboxCL.original.dropbox.autostart` (*argv*)

automatically start dropbox at login dropbox autostart [y/n]

options: n dropbox will not start automatically at login y dropbox will start automatically at login (default)

Note: May only work on current Ubuntu distributions.

`dryxDropboxCL.original.dropbox.exclude` (**n, **kw*)

ignores/excludes a directory from syncing dropbox exclude [list] dropbox exclude add [DIRECTORY] [DIRECTORY] ... dropbox exclude remove [DIRECTORY] [DIRECTORY] ...

“list” prints a list of directories currently excluded from syncing. “add” adds one or more directories to the exclusion list, then resynchronizes Dropbox. “remove” removes one or more directories from the exclusion list, then resynchronizes Dropbox. With no arguments, executes “list”. Any specified path must be within Dropbox.

`dryxDropboxCL.original.dropbox.filestatus` (**n, **kw*)

get current sync status of one or more files dropbox filestatus [-l] [-a] [FILE]...

Prints the current status of each FILE.

options: -l –list prints out information in a format similar to ls. works best when your console supports color :)
-a –all do not ignore entries starting with .

Aliases: stat

`dryxDropboxCL.original.dropbox.help` (*argv*)

provide help dropbox help [COMMAND]

With no arguments, print a list of commands and a short description of each. With a command, print descriptive help on how to use the command.

`dryxDropboxCL.original.dropbox.lansync` (**n, **kw*)

enables or disables LAN sync dropbox lansync [y/n]

options: y dropbox will use LAN sync (default) n dropbox will not use LAN sync

`dryxDropboxCL.original.dropbox.ls` (**n, **kw*)

list directory contents with current sync status dropbox ls [FILE]...

This is an alias for filestatus -l

`dryxDropboxCL.original.dropbox.puburl` (**n, **kw*)

get public url of a file in your dropbox dropbox puburl FILE

Prints out a public url for FILE.

`dryxDropboxCL.original.dropbox.relpath (path, start='.')`
Return a relative version of a path

`dryxDropboxCL.original.dropbox.running (argv)`
 return whether dropbox is running dropbox running
 Returns 1 if running 0 if not running.

`dryxDropboxCL.original.dropbox.start (argv)`
 start dropboxd dropbox start [-i]
 Starts the dropbox daemon, dropboxd. If dropboxd is already running, this will do nothing.
options: -i --install auto install dropboxd if not available on the system

`dryxDropboxCL.original.dropbox.status (*n, **kw)`
 get current status of the dropboxd dropbox status
 Prints out the current status of the Dropbox daemon.

`dryxDropboxCL.original.dropbox.stop (*n, **kw)`
 stop dropboxd dropbox stop
 Stops the dropbox daemon.

8.3 Classes

<code>dryxDropboxCL.utKit.utKit</code>	<i>Override dryx utKit</i>
<code>dryxDropboxCL.original.dropbox.CommandTicker</code>	
<code>dryxDropboxCL.original.dropbox.DownloadState</code>	
<code>dryxDropboxCL.original.dropbox.DropboxCommand</code>	
<code>dryxDropboxCL.original.dropbox.SignatureVerifyError</code>	

8.3.1 dryxDropboxCL.utKit.utKit (class)

class `dryxDropboxCL.utKit.utKit (moduleDirectory)`
Override dryx utKit
__init__ (moduleDirectory)

Methods

<code>__init__ (moduleDirectory)</code>	
<code>setupModule()</code>	<i>The setupModule method</i>
<code>tearDownModule()</code>	<i>The tearDownModule method</i>

8.3.2 dryxDropboxCL.original.dropbox.CommandTicker (class)

class `dryxDropboxCL.original.dropbox.CommandTicker`
__init__ ()

Methods

<code>__init__()</code>	
<code>getName()</code>	
<code>isAlive()</code>	Return whether the thread is alive.
<code>isDaemon()</code>	
<code>is_alive()</code>	Return whether the thread is alive.
<code>join([timeout])</code>	Wait until the thread terminates.
<code>run()</code>	
<code>setDaemon(daemonic)</code>	
<code>setName(name)</code>	
<code>start()</code>	Start the thread's activity.
<code>stop()</code>	

Attributes

<code>daemon</code>	A boolean value indicating whether this thread is a daemon thread (True) or not (False).
<code>ident</code>	Thread identifier of this thread or None if it has not been started.
<code>name</code>	A string used for identification purposes only.

8.3.3 `dryxDropboxCL.original.dropbox.DownloadState` (*class*)

class `dryxDropboxCL.original.dropbox.DownloadState`

`__init__()`

Methods

<code>__init__()</code>
<code>cancel()</code>
<code>copy_data()</code>
<code>unpack()</code>

8.3.4 `dryxDropboxCL.original.dropbox.DropboxCommand` (*class*)

class `dryxDropboxCL.original.dropbox.DropboxCommand` (*timeout=5*)

`__init__(timeout=5)`

Methods

<code>__init__([timeout])</code>
<code>close()</code>
<code>send_command(name, args)</code>

8.3.5 dryxDropboxCL.original.dropbox.SignatureVerifyError (function)

exception `dryxDropboxCL.original.dropbox.SignatureVerifyError`

8.4 Functions

<code>dryxDropboxCL.setup_selective_sync</code>	<i>setup selective sync</i>
<code>dryxDropboxCL.start_if_not_running</code>	<i>start if not running</i>
<code>dryxDropboxCL.original.dropbox.alias</code>	
<code>dryxDropboxCL.original.dropbox.autostart</code>	automatically start dropbox at login
<code>dryxDropboxCL.original.dropbox.can_reroll_autostart</code>	
<code>dryxDropboxCL.original.dropbox.columnize</code>	
<code>dryxDropboxCL.original.dropbox.command</code>	
<code>dryxDropboxCL.original.dropbox.console_flush</code>	
<code>dryxDropboxCL.original.dropbox.console_print</code>	
<code>dryxDropboxCL.original.dropbox.download</code>	
<code>dryxDropboxCL.original.dropbox.download_file_chunk</code>	
<code>dryxDropboxCL.original.dropbox.exclude</code>	ignores/excludes a directory from syncing
<code>dryxDropboxCL.original.dropbox.filestatus</code>	get current sync status of one or more files
<code>dryxDropboxCL.original.dropbox.gpgme_context</code>	
<code>dryxDropboxCL.original.dropbox.grab_link_url_if_necessary</code>	
<code>dryxDropboxCL.original.dropbox.help</code>	provide help
<code>dryxDropboxCL.original.dropbox.is_dropbox_running</code>	
<code>dryxDropboxCL.original.dropbox.lansync</code>	enables or disables LAN sync
<code>dryxDropboxCL.original.dropbox.load_serialized_images</code>	
<code>dryxDropboxCL.original.dropbox.ls</code>	list directory contents with current sync status
<code>dryxDropboxCL.original.dropbox.methodcaller</code>	
<code>dryxDropboxCL.original.dropbox.plat</code>	
<code>dryxDropboxCL.original.dropbox.puburl</code>	get public url of a file in your dropbox
<code>dryxDropboxCL.original.dropbox.relpeth</code>	<i>Return a relative version of a path</i>
<code>dryxDropboxCL.original.dropbox.requires_dropbox_running</code>	
<code>dryxDropboxCL.original.dropbox.reroll_autostart</code>	
<code>dryxDropboxCL.original.dropbox.running</code>	return whether dropbox is running
<code>dryxDropboxCL.original.dropbox.start</code>	start dropboxd
<code>dryxDropboxCL.original.dropbox.start_dropbox</code>	
<code>dryxDropboxCL.original.dropbox.status</code>	get current status of the dropboxd
<code>dryxDropboxCL.original.dropbox.stop</code>	stop dropboxd
<code>dryxDropboxCL.original.dropbox.unicode_abspath</code>	
<code>dryxDropboxCL.original.dropbox.usage</code>	
<code>dryxDropboxCL.original.dropbox.verify_signature</code>	
<code>dryxDropboxCL.original.dropbox.yes_no_question</code>	

8.4.1 dryxDropboxCL.setup_selective_sync (function)

`dryxDropboxCL.setup_selective_sync` (*log*, *settings*)
setup selective sync

Key Arguments:

- *log* – logger

Return:

- None

Todo

- @review: when complete, clean setup_selective_sync function
 - @review: when complete add logging
 - @review: when complete, decide whether to abstract function to another module
-

8.4.2 dryxDropboxCL.start_if_not_running (*function*)

`dryxDropboxCL.start_if_not_running(log)`
start if not running

Key Arguments:

- log – logger

Return:

- None

Todo

- @review: when complete, clean start_if_not_running function
 - @review: when complete add logging
 - @review: when complete, decide whether to abstract function to another module
-

8.4.3 dryxDropboxCL.original.dropbox.alias (*function*)

`dryxDropboxCL.original.dropbox.alias(name)`

8.4.4 dryxDropboxCL.original.dropbox.autostart (*function*)

`dryxDropboxCL.original.dropbox.autostart(argv)`
automatically start dropbox at login dropbox autostart [y/n]

options: n dropbox will not start automatically at login y dropbox will start automatically at login (default)

Note: May only work on current Ubuntu distributions.

8.4.5 dryxDropboxCL.original.dropbox.can_reroll_autostart (*function*)

`dryxDropboxCL.original.dropbox.can_reroll_autostart()`

8.4.6 dryxDropboxCL.original.dropbox.columnize (*function*)

`dryxDropboxCL.original.dropbox.columnize(list, display_list=None, display_width=None)`

8.4.7 dryxDropboxCL.original.dropbox.command (*function*)

`dryxDropboxCL.original.dropbox.command (meth)`

8.4.8 dryxDropboxCL.original.dropbox.console_flush (*function*)

`dryxDropboxCL.original.dropbox.console_flush (f=<open file '<stdout>', mode 'w'>)`

8.4.9 dryxDropboxCL.original.dropbox.console_print (*function*)

`dryxDropboxCL.original.dropbox.console_print (st='u', f=<open file '<stdout>', mode 'w'>, linebreak=True)`

8.4.10 dryxDropboxCL.original.dropbox.download (*function*)

`dryxDropboxCL.original.dropbox.download ()`

8.4.11 dryxDropboxCL.original.dropbox.download_file_chunk (*function*)

`dryxDropboxCL.original.dropbox.download_file_chunk (url, buf)`

8.4.12 dryxDropboxCL.original.dropbox.exclude (*function*)

`dryxDropboxCL.original.dropbox.exclude (*n, **kw)`

ignores/excludes a directory from syncing dropbox exclude [list] dropbox exclude add [DIRECTORY] [DIRECTORY] ... dropbox exclude remove [DIRECTORY] [DIRECTORY] ...

“list” prints a list of directories currently excluded from syncing. “add” adds one or more directories to the exclusion list, then resynchronizes Dropbox. “remove” removes one or more directories from the exclusion list, then resynchronizes Dropbox. With no arguments, executes “list”. Any specified path must be within Dropbox.

8.4.13 dryxDropboxCL.original.dropbox.filestatus (*function*)

`dryxDropboxCL.original.dropbox.filestatus (*n, **kw)`

get current sync status of one or more files dropbox filestatus [-l] [-a] [FILE]...

Prints the current status of each FILE.

options: -l –list prints out information in a format similar to ls. works best when your console supports color :)
-a –all do not ignore entries starting with .

Aliases: stat

8.4.14 dryxDropboxCL.original.dropbox.gpgme_context (*function*)

`dryxDropboxCL.original.dropbox.gpgme_context (*args, **kwds)`

8.4.15 `dryxDropboxCL.original.dropbox.grab_link_url_if_necessary` (*function*)

```
dryxDropboxCL.original.dropbox.grab_link_url_if_necessary()
```

8.4.16 `dryxDropboxCL.original.dropbox.help` (*function*)

```
dryxDropboxCL.original.dropbox.help(argv)
provide help dropbox help [COMMAND]
```

With no arguments, print a list of commands and a short description of each. With a command, print descriptive help on how to use the command.

8.4.17 `dryxDropboxCL.original.dropbox.is_dropbox_running` (*function*)

```
dryxDropboxCL.original.dropbox.is_dropbox_running()
```

8.4.18 `dryxDropboxCL.original.dropbox.lansync` (*function*)

```
dryxDropboxCL.original.dropbox.lansync(*n, **kw)
enables or disables LAN sync dropbox lansync [y/n]
```

options: y dropbox will use LAN sync (default) n dropbox will not use LAN sync

8.4.19 `dryxDropboxCL.original.dropbox.load_serialized_images` (*function*)

```
dryxDropboxCL.original.dropbox.load_serialized_images()
```

8.4.20 `dryxDropboxCL.original.dropbox.ls` (*function*)

```
dryxDropboxCL.original.dropbox.ls(*n, **kw)
list directory contents with current sync status dropbox ls [FILE]...
```

This is an alias for `filestatus -l`

8.4.21 `dryxDropboxCL.original.dropbox.methodcaller` (*function*)

```
dryxDropboxCL.original.dropbox.methodcaller(name, *args, **kwargs)
```

8.4.22 `dryxDropboxCL.original.dropbox.plat` (*function*)

```
dryxDropboxCL.original.dropbox.plat()
```

8.4.23 `dryxDropboxCL.original.dropbox.puburl` (*function*)

```
dryxDropboxCL.original.dropbox.puburl(*n, **kw)
get public url of a file in your dropbox dropbox puburl FILE
```

Prints out a public url for FILE.

8.4.24 `dryxDropboxCL.original.dropbox.relpath` (*function*)

`dryxDropboxCL.original.dropbox.relpath` (*path*, *start*='.')

Return a relative version of a path

8.4.25 `dryxDropboxCL.original.dropbox.requires_dropbox_running` (*function*)

`dryxDropboxCL.original.dropbox.requires_dropbox_running` (*meth*)

8.4.26 `dryxDropboxCL.original.dropbox.reroll_autostart` (*function*)

`dryxDropboxCL.original.dropbox.reroll_autostart` (*should_autostart*)

8.4.27 `dryxDropboxCL.original.dropbox.running` (*function*)

`dryxDropboxCL.original.dropbox.running` (*argv*)

return whether dropbox is running dropbox running

Returns 1 if running 0 if not running.

8.4.28 `dryxDropboxCL.original.dropbox.start` (*function*)

`dryxDropboxCL.original.dropbox.start` (*argv*)

start dropboxd dropbox start [-i]

Starts the dropbox daemon, dropboxd. If dropboxd is already running, this will do nothing.

options: -i --install auto install dropboxd if not available on the system

8.4.29 `dryxDropboxCL.original.dropbox.start_dropbox` (*function*)

`dryxDropboxCL.original.dropbox.start_dropbox` ()

8.4.30 `dryxDropboxCL.original.dropbox.status` (*function*)

`dryxDropboxCL.original.dropbox.status` (**n*, ***kw*)

get current status of the dropboxd dropbox status

Prints out the current status of the Dropbox daemon.

8.4.31 `dryxDropboxCL.original.dropbox.stop` (*function*)

`dryxDropboxCL.original.dropbox.stop` (**n*, ***kw*)

stop dropboxd dropbox stop

Stops the dropbox daemon.

8.4.32 `dryxDropboxCL.original.dropbox.unicode_abbrev` (*function*)

`dryxDropboxCL.original.dropbox.unicode_abbrev` (*path*)

8.4.33 `dryxDropboxCL.original.dropbox.usage` (*function*)

`dryxDropboxCL.original.dropbox.usage` (*argv*)

8.4.34 `dryxDropboxCL.original.dropbox.verify_signature` (*function*)

`dryxDropboxCL.original.dropbox.verify_signature` (*key_file, sig_file, plain_file*)

8.4.35 `dryxDropboxCL.original.dropbox.yes_no_question` (*function*)

`dryxDropboxCL.original.dropbox.yes_no_question` (*question*)

8.5 Indexes

- [Module Index](#)
- [Full Index](#)

8.6 Todo

- [Todolist](#)

d

`dryxDropboxCL`, [17](#)
`dryxDropboxCL.cl_utils`, [17](#)
`dryxDropboxCL.commonutils`, [17](#)
`dryxDropboxCL.original`, [17](#)
`dryxDropboxCL.original.dropbox`, [18](#)
`dryxDropboxCL.utKit`, [18](#)

Symbols

- `__init__()` (dryxDropboxCL.original.dropbox.CommandTickerView method), 19
 - `__init__()` (dryxDropboxCL.original.dropbox.DownloadState method), 20
 - `__init__()` (dryxDropboxCL.original.dropbox.DropboxCommand method), 20
 - `__init__()` (dryxDropboxCL.utKit.utKit method), 19
- ## A
- `alias()` (in module dryxDropboxCL.original.dropbox), 22
 - `autostart()` (in module dryxDropboxCL.original.dropbox), 18, 22
- ## C
- `can_reroll_autostart()` (in module dryxDropboxCL.original.dropbox), 22
 - `columnize()` (in module dryxDropboxCL.original.dropbox), 22
 - `command()` (in module dryxDropboxCL.original.dropbox), 23
 - `CommandTickerView` (class in dryxDropboxCL.original.dropbox), 19
 - `console_flush()` (in module dryxDropboxCL.original.dropbox), 23
 - `console_print()` (in module dryxDropboxCL.original.dropbox), 23
- ## D
- `download()` (in module dryxDropboxCL.original.dropbox), 23
 - `download_file_chunk()` (in module dryxDropboxCL.original.dropbox), 23
 - `DownloadState` (class in dryxDropboxCL.original.dropbox), 20
 - `DropboxCommand` (class in dryxDropboxCL.original.dropbox), 20
 - `dryxDropboxCL` (module), 17
 - `dryxDropboxCL.cl_utils` (module), 17
 - `dryxDropboxCL.commonutils` (module), 17
 - `dryxDropboxCL.original` (module), 17
 - `dryxDropboxCL.original.dropbox` (module), 18
 - `dryxDropboxCL.utKit` (module), 18
- ## E
- `exclude()` (in module dryxDropboxCL.original.dropbox), 18, 23
- ## F
- `filestatus()` (in module dryxDropboxCL.original.dropbox), 18, 23
- ## G
- `gpgme_context()` (in module dryxDropboxCL.original.dropbox), 23
 - `grab_link_url_if_necessary()` (in module dryxDropboxCL.original.dropbox), 24
- ## H
- `help()` (in module dryxDropboxCL.original.dropbox), 18, 24
- ## I
- `is_dropbox_running()` (in module dryxDropboxCL.original.dropbox), 24
- ## L
- `lansync()` (in module dryxDropboxCL.original.dropbox), 18, 24
 - `load_serialized_images()` (in module dryxDropboxCL.original.dropbox), 24
 - `ls()` (in module dryxDropboxCL.original.dropbox), 18, 24
- ## M
- `main()` (in module dryxDropboxCL.cl_utils), 18
 - `methodcaller()` (in module dryxDropboxCL.original.dropbox), 24
- ## P
- `plat()` (in module dryxDropboxCL.original.dropbox), 24

puburl() (in module dryxDropboxCL.original.dropbox),
[18](#), [24](#)

R

relpath() (in module dryxDropboxCL.original.dropbox),
[18](#), [25](#)

requires_dropbox_running() (in module dryxDropboxCL.original.dropbox), [25](#)

reroll_autostart() (in module dryxDropboxCL.original.dropbox), [25](#)

running() (in module dryxDropboxCL.original.dropbox),
[19](#), [25](#)

S

setup_selective_sync() (in module dryxDropboxCL), [21](#)

SignatureVerifyError, [21](#)

start() (in module dryxDropboxCL.original.dropbox), [19](#),
[25](#)

start_dropbox() (in module dryxDropboxCL.original.dropbox), [25](#)

start_if_not_running() (in module dryxDropboxCL), [22](#)

status() (in module dryxDropboxCL.original.dropbox),
[19](#), [25](#)

stop() (in module dryxDropboxCL.original.dropbox), [19](#),
[25](#)

U

unicode_abspath() (in module dryxDropboxCL.original.dropbox), [25](#)

usage() (in module dryxDropboxCL.original.dropbox),
[26](#)

utKit (class in dryxDropboxCL.utKit), [18](#), [19](#)

V

verify_signature() (in module dryxDropboxCL.original.dropbox), [26](#)

Y

yes_no_question() (in module dryxDropboxCL.original.dropbox), [26](#)