

# Home

2018-12-20: Version 0.5 released.

## Disclaimer

As with any copy/sync tool, make sure to test thoroughly with data that can be lost.

Test with the `--dry` and `--dry-rsync` options to verify that the operations match what you want to do.

## Contents

- 1 Introduction
- 2 Why not just use `rsync` directly?
- 3 Usage
- 4 Configuration
- 5 Examples
- 6 Mirroring Selected File Objects

## Introduction

`rmirr` provides a **safe** and easy to use way to set up and execute file storage mirroring instructions. And because it uses `rsync` to actually perform the synchronization, you get all the performance and availability benefits of `rsync`.

Features:

- specify only the source file object (file, directory) to mirror
- mirror settings selected by best match path
- support one or more mirror destinations
- ensure mirroring from specific sources only
- ensure single instance of mirroring operation per mirror path
- log history of mirroring invocations
- save to file report of mirrored objects
- email report file

## Why not just use `rsync` directly?

Simple cases of mirroring files/directories can easily be done by using `rsync` directly. But, it does not provide any safeguards. For example, the following are not the same:

```
rsync -avz --delete /a/b/c mach2:/a/b/c
```

and:

```
rsync -avz --delete /a/b/c/ mach2:/a/b/c
```

Assuming a directory is being synced, call #2 is correct, as it uses a trailing `/` on the source path. `rmirr` ensures that will be done properly.

## Usage

## About

### Name

`rmirr` - Mirroring Tool

### Requirements

Python 2, `rsync`

### License

GPL-2

### Links

[Repository](#)

## Featured

### Recently Updated

As you and your team create content this area will fill up and display the latest updates.

```
usage: rmirr [<options>] [-n <name>] -p <path>
       rmirr [<options>] [-n <name>] -s
<suitename>
       rmirr [<options>] -n <name>
       rmirr -l
```

Mirror file objects which can be specified by <path> (for which a mirror is selected), <suite> (representing one or more paths), or <name> (a mirror configuration).

Where:

```
-l      List mirror configurations.
-n <name>
       Mirror name.
-p <path>
       Mirror path.
-s <suitename>
       Alias for one or more paths.
```

Options:

```
-c <path>
       Location of paths.json configuration
file.
-d <hostname>[,...]
       Destinations to mirror to from list in
configuration.
--debug
       Enable debugging.
--delete
       Allow file object deletion.
--dry   Dry run. Do not execute.
--dry-rsync
       Dry run for rsync.
--mailto <emailaddr>[,...]
       Set/override recipients when mailing
report.
--mailreport
       Mail report.
--nolock
       Do not use/require lock to run.
--safeoff
       Disable safemode.
--showreport
       Show report on console.
--verbose
       Enable verbosity.
-y      Do not ask for confirmation before
executing.
```

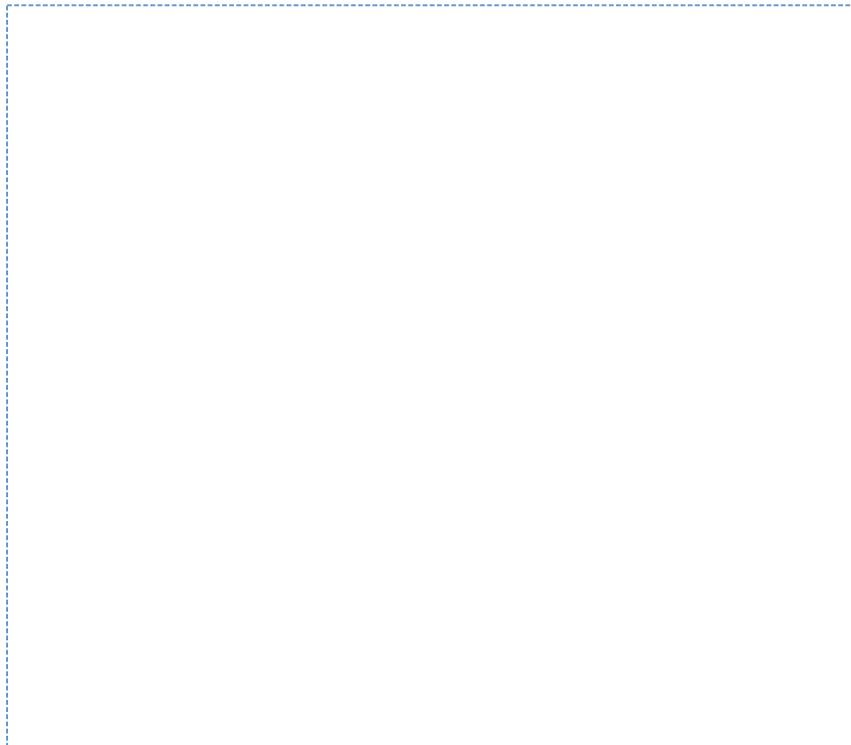
# Configuration

Configuration file is located at `~/ .rmirr/rmirr.json`.

Section	Name	Description
defaults	email_recipients	List of email recipients when <code>--mailreport</code> is used.
	smtp_server	Set SMTP server. Default is to use "localhost".
suites	<suitename>	List of paths to be mirrored.
mirrors	comment	Description of what is being mirrored.
	destinations	List of destinations to which the source is to be mirrored each of the format [ <code>&lt;user&gt;@&lt;host&gt;:&lt;path&gt;</code> ].
	email_recipients	List of email recipients when <code>--mailreport</code> is used. Overrides the value from the defaults section.
	excludes	List of file/dir names to exclude from the mirroring.
	name	Mirror configuration name.
	source	Source of the mirror: [ <code>&lt;user&gt;@&lt;host&gt;:&lt;path&gt;</code> ].

# Examples

Given the configuration at `~/ .rmirr/rmirr.json`:



```

{
  "defaults": {
    "email_recipients": []
  },
  "suites": {
    "all": [
      "/data/main",
      "/data/finance",
      "/data/hr"
    ]
  },
  "mirrors": [
    {
      "name": "data-main--
home--finance-hr",
      "comment": "data main:
home -> finance, hr",
      "source": "data@home.
abc.xyz:/data/main",
      "excludes": [".
snapshot", ".snapshots"],
      "destinations":
["finance.abc.xyz", "hr.abc.xyz"]
    },
    {
      "name": "data-finance--
home--finance",
      "comment": "data
finance: home -> finance",
      "email_recipients":
["bob@finance.abc.xyz"],
      "source": "data@home.
abc.xyz:/data/finance",
      "excludes": [".
snapshot", ".snapshots"],
      "destinations":
["finance.abc.xyz"]
    },
    {
      "name": "data-hr--
home--hr",
      "comment": "data hr:
home -> hr",
      "source": "data@home.
abc.xyz:/data/hr",
      "excludes": [".
snapshot", ".snapshots"],
      "destinations": ["hr.
abc.xyz"]
    }
  ]
}

```

List suite and mirror settings:

```
rmirr -l
```

Mirror /data/main from home.abc.xyz:

```
rmirr -p /data/main
```

Notes:

- mirrors to both `finance.abc.xyz` and `hr.abc.xyz`
- `.snapshot` and `.snapshots` are excluded
- by default, mirroring **does not** delete at the destination
- by default, mirroring **requires** user interaction ("y" to proceed)

Show what `rmirr` would do **without** executing it (dry-run):

```
rmirr --dry -p /data/main
```

Mirror /data/main/opt:

```
rmirr -p /data/main/opt
```

Notes:

- because `/data/main/opt` is under `/data/main`, the settings for `/data/main` are used
- only `/data/main/opt` is mirrored

Mirror /data/finance and allow deletion at the destination:

```
rmirr --delete -p /data/finance
```

Do a dry run (at `rsync` level) to mirror /data/finance with deletion and no user interaction:

```
rmirr --dry-rsync --delete -y -p /data/finance
```

Mirror /data/main to specific destination (`hr.abc.xyz`):

```
rmirr --delete -y -d hr.abc.xyz -p /data/main
```

Request report be emailed:

```
rmirr -p /data/main --mailreport
```

Request report be output to console:

```
rmirr -p /data/main --showreport
```

Mirror the "all" suite:

```
rmirr -s all
```

## Mirroring Selected File Objects

Sometimes only a subset of file objects (files, directories) need to be mirrored rather than a full tree. `rmirr` supports this with the `names` setting in the configuration file.

Configuration for mirroring bash files:

```
{
  "suites": {
    "bashfiles": [
      "~/bash_completion.d",
      "~/bash_profile",
      "~/bash_logout",
      "~/bashrc",
      ~/.profile",
      ~/.profile.d"
    ]
  },
  "mirrors": [
    {
      "name": "my-bash-
files",
      "comment": "my .bash
files",
      "source": "jdm@home.
abc.xyz:~",
      "excludes": [
        ".snapshot",
        ".snapshots"
      ],
      "destinations": [
        "finance.abc.
xyz",
        "hr.abc.xyz"
      ]
    }
  ]
}
```

Notes:

- the `suites` section defines a suite to reference those bash-specific file objects by a single name

- of course, any changes/tweaks done at the destinations will be overwritten when `rmirr` is called

To mirror all the file objects of the "bashfiles" suite:

```
rmirr -s bashfiles
```

To mirror specific file objects:

```
rmirr -p /home/jdm/.bash_completion.d
```